(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 December 2000 (07.12.2000)

PCT

(10) International Publication Number
WO 00/74367 A2

(51) International Patent Classification[7]: H04N

(21) International Application Number: PCT/US00/14866

(22) International Filing Date: 31 May 2000 (31.05.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/322,976 1 June 1999 (01.06.1999) US

(71) Applicant (*for all designated States except US*): TECHNION RESEARCH AND DEVELOPMENT FOUNDATION LTD. [IL/IL]; Technion City, 32000 Haifa (IL).

(71) Applicant (*for TJ only*): FRIEDMAN, Mark, M. [US/IL]; 1 Alharizi St., 43406 Raanana (IL).

(72) Inventors; and
(75) Inventors/Applicants (*for US only*): BIRK, Yitzhak [IL/IL]; 33 Anshei Breishit St., 45267 Hod Hasharon (IL). MONDRI, Ron [IL/IL]; 11 Vitkin St., 34756 Haifa (IL).

(74) Common Representative: FRIEDMAN, Mark, M.; c/o Castorina, Anthony, 2001 Jefferson Davis Highway, Suite 207, Arlington, VA 22202 (US).

(81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD OF OPTIMIZING NEAR-VIDEO-ON-DEMAND TRANSMISSION

(57) Abstract: A system and method for Near Video On Demand (NVOD) transmission of a program from a server to a plurality of clients. The program is partitioned into segments. Each segment is transmitted repeatedly, with the transmission rate of segments subsequent to the first segment being lower than the transmission rate of the first segment. The transmission of the segments is effected in a manner that minimizes the aggregate transmission bandwidth, subject to constraints related to client parameters such as client storage capacity and client recording rates. Preferably, the sequences are partitioned further into subsequences, and redundant subsequences are used for error correction. Preferably, the segments, or the subsegments, include metadata such as segment or subsegment length, segment or subsegment sequence number, or the time until the next transmission of the sequence or subsequence. The segments include metadata that describe the transmission plan, enabling a client to pause the display of the program at any point and resume the display at will. A "live" program may be partitioned into segments that are transmitted concurrently with the live program, giving clients the option of toggling between displaying the program in progress and displaying the program in its entirety.

WO 00/74367 A2

METHOD OF OPTIMIZING NEAR-VIDEO-ON-DEMAND TRANSMISSION

FIELD AND BACKGROUND OF THE INVENTION

The present invention relates to a method and system for transmitting a program to a plurality of viewers, permitting flexible, nearly on demand viewing by an unlimited number of concurrent viewers, and, more particularly, to a method and system for transmission that minimizes transmission bandwidth, possibly subject to certain constraints.

Video-On-Demand (VOD) is the on-line version of traditional video-rental services. As with video rental services, each viewer receives a dedicated "copy" of the movie and can view it in a flexible manner, including the ability to pause and resume, rewind, and possibly even fast-forward. With VOD, the "rental" operation is essentially instantaneous, and viewing can begin within seconds of the decision to view.

The "copy" in VOD is a dedicated video stream. This stream is generated by a video server and sent to the viewer over a communication network. An important advantage of VOD over tape rental is the great flexibility in allocation of resources: the maximum number of concurrent viewers is independent of viewing choices, and is limited only by the server's total streaming capacity. The required bandwidth resources, both in the server and in the communication network, are proportional to the number of concurrent viewers.

There are important situations in which a large number of people wish to view the same content during the same period of time, albeit not simultaneously. One example is viewing emergency-preparedness instruction in the hours or days prior to the arrival of a major storm. Another example is a newly released "hot" movie that is advertised heavily. Yet another example is a movie whose viewing is assigned as homework, or even a recorded lecture viewed (individually) in class by the students. These pre-recorded instructions, movies and lectures are examples of programs that are to be viewed by many viewers, concurrently but not necessarily simultaneously.

Although VOD could be used to address such situations, it is both highly desirable and intuitively possible to do better. The desire stems from the fact that even the total (over all programs) number of concurrent viewers may be temporarily

much higher than usual, so it would be very costly if not impossible to design the infrastructure (server and communication network) for such peaks. The intuition that something can be done arises from the observation that the many viewers of the "hot" program are viewing the same material concurrently but not simultaneously. The various schemes for doing better than VOD in this situation are called "*Near Video On Demand*". The goal of NVOD is to provide an unlimited number of viewers of the same program similar service flexibility to that of VOD at a reasonable cost to the server and communication network. Ideally, this cost is independent of the number of viewers. "Near" is defined to mean commencement of viewing within a reasonable time interval following viewer request, for example one minute in the case of a movie, as well as the ability to pause and resume at any time. Rewind and fast-forward functions are not obligatory.

There are two categories of NVOD systems: open-loop systems and closed-loop systems. In both systems, the viewers are provided with devices, called herein "clients" because of their relationship with the server, that receive program copies transmitted by the server and display those copies to their respective viewers. In open-loop systems there is no feedback from the viewing client to the server, so neither server transmissions nor routing on the network are affected by viewer actions (other than the possible effect on routing due to a viewer joining a multicast group). Open-loop schemes lend themselves most naturally to broadcast-based networks, such as cable television networks, and even to networks that have only one-way communication, which is the common case in satellite-based information-dissemination networks. Closed-loop systems permit some feedback that allows the server to adjust to client requests throughout the viewing period. Note that the terms "play" and "display" are used interchangeably herein, to refer to the displaying of the received program by a client.

Recently, several open-loop NVOD schemes have been proposed. These schemes are based on partitioning the program into several segments, on the assumption that every client has a substantial amount of available storage capacity, for example on a hard disk, which can be used to temporarily store the segments. In such schemes, the server's transmission schedule, and the algorithm used by the client to decide whether or not to record any given transmitted segment, jointly ensure that

every segment of the movie is stored in the client's recording medium by its viewing time.

One such scheme is taught by DeBey in U. S. Patent No. 5,421,031, which is incorporated by reference for all purposes as if fully set forth herein. DeBey's partitioning and scheduling scheme is illustrated in Figure 1, for the case of segments of equal length. The vertical axis of Figure 1 is segment number. The horizontal axis of Figure 1 is the time at which a given segment is broadcast by the server, with the unit of time, as well as the basic time interval, being the duration of one segment. For each segment, the time during which that segment is broadcast by the server is represented by a double-headed arrow. The first segment is broadcast in every time interval, the second segment is broadcast every second time interval, the third segment is broadcast every third time interval, and in general the n-th segment is broadcast every n-th time interval. Note that all segments are transmitted at the same transmission rate of one segment per time interval. In addition, the transmissions continue throughout the time period during which the viewers are permitted to view the program.

A client that tunes in to the broadcast at the beginning of any time interval receives all the segments promptly enough to display the program with no interruptions. For example, a client that tunes in at the beginning of the seventh time interval, and that actually begins to display the movie to its viewer at the beginning of the eighth time interval, receives and records the seventh copy of the first segment during the seventh time interval, the fourth copy of the second segment during the eighth time interval, the third copy of the third segment during the ninth time interval, the second copy of the fourth segment during the eighth time interval, etc. In this case, the first segment is displayed during the eighth time interval, the second segment is displayed during the ninth time interval, the third segment is displayed during the tenth time interval, the fourth segment is displayed during the eleventh time interval, etc.

DeBey's scheme imposes certain burdens on the server and on the clients. With N segments, the mean transmission bandwidth, in units of segments transmitted per time interval, is approximately $\ln(N)$; but the actual transmission bandwidth varies widely. For example, in prime-numbered time intervals after the first time interval,

only two segments are broadcast, vs. *e.g.* six segments during the twelfth time interval. The client must be able to record the received segments fast enough to keep up with the peak aggregate transmission rate. Furthermore, the client must have enough storage capacity to store all recorded segments that are received too soon to play.

There is thus a widely recognized need for, and it would be highly advantageous to have, a NVOD method that imposes less of a burden on the resources available to the server and to the clients.

SUMMARY OF THE INVENTION

According to the present invention there is provided, in a system wherein a server transmits a program having a certain duration, the program being received by at least one client, a method for planning the transmission of the program, including the steps of: (a) partitioning the program into a plurality of sequential segments; and (b) selecting a transmission rate for each segment, the transmission rate that is selected for a first segment being faster than the transmission rate that is selected for any other segment.

According to the present invention there is provided a system for transmitting a program to at least one viewer, including: (a) a software module including a plurality of instructions for transmitting the program by: (i) partitioning the program into a plurality of sequential segments, and (ii) selecting a transmission rate for each segment, the transmission rate that is selected for a first the segment being faster than the transmission rate that is selected for any other the segment; (b) a processor for executing the instructions; (c) a server for transmitting each segment at the respective transmission rate; and (d) for each at least one viewer, a client for receiving the transmitted segments, recording the received segments and playing the recorded segments in the sequence.

According to the present invention there is provided, in a system wherein a server transmits a program that is partitioned into a plurality of segments, the segments being transmitted repeatedly, and wherein a client receives and records the segments and displays the program, the transmitting, receiving and recording of the segments being effected according to a transmission plan, a method for displaying the

program intermittently, including the steps of: (a) transmitting, along with the segments, metadata describing the transmission plan, by the server; (b) pausing the display of the program, by the client; (c) resuming the display of the program, by the client, subsequent to the pausing; and (d) during the pausing, continuing to record at least a portion of the segments then received, by the client.

According to the present invention there is provided, in a system wherein a server transmits a program that is received by at least one client, a method for planning the transmission of the program, including the steps of: (a) partitioning the program into a plurality of sequential segments; and (b) selecting a transmission rate for each segment, the selecting being effected in a manner that substantially minimizes a total transmission bandwidth, subject to a constraint based on at least one parameter of the at least one client.

According to the present invention there is provided a system for transmitting a program to at least one viewer, including: (a) a software module including a plurality of instructions for transmitting the program by: (i) partitioning the program into a plurality of sequential segments, and (ii) selecting a transmission rate for each segment; (b) a processor for executing the instructions; (c) a server for transmitting each segment repeatedly at the respective transmission rate; and (d) for each at least one viewer, a client for receiving the transmitted segments, recording the received segments and playing the recorded segments in the sequence; and wherein the transmission rates are selected in a manner that substantially minimizes a total transmission bandwidth, subject to a constraint based on at least one parameter of the at least one client.

According to the present invention there is provided a method for transmitting a program from a server to at least one client, including the steps of: (a) partitioning the program into a plurality of sequential segments; (b) selecting a transmission rate for each segment; (c) transmitting the program as a single unit, by the server; (d) transmitting the segments, by the server, each segment being transmitted at the respective transmission rate, the transmitting of at least a portion of the segments being concurrent with the transmitting of the single unit; and (e) for one of the at least one client: (i) receiving both the transmitted segments and the transmitted single unit, subsequent to a start of the transmitting of the single unit, so that the one client

receives only a portion of the single unit, and (ii) effecting a step selected from the group consisting of: (A) displaying the portion of the single unit, and (B) displaying the segments in the sequence, starting from the first segment.

According to the present invention there is provided a method for transmitting a plurality of programs from a server to at least one client, including the steps of: (a) partitioning each program into a plurality of sequential segments; (b) for each program, selecting a transmission rate for each segment; (c) transmitting the programs sequentially, by the server, each program being transmitted as a single unit; and (d) for each program, transmitting the segments of the program, by the server, each segment being transmitted at the respective transmission rate, the transmitting of at least a portion of the segments being concurrent with the transmitting of the program as a single unit.

The method of the present invention is referred herein as a method of "planning" the transmission of a program, because the method of the present invention determines the data rate at which each segment is to be transmitted. Unlike in certain prior art, for example that taught by DeBey, there is no need to determine in advance the transmission schedule, namely the absolute or even the relative times at which different must be transmitted. Thus, there is no need for a predetermined transmission schedule, although such a predetermined transmission schedule could be constructed. According to the present invention, the actual transmission of a program simply entails the substantially concurrent repeated transmission of the program segments at the respective data rates assigned to them by the method of the present invention. Nevertheless and without loss of generality or implication of the need for or existence of a predetermined transmission schedule, the actual transmission rates that are determined for the various segments are sometimes referred to herein as a "transmission schedule".

The determination of transmission rates for the various segments may furthermore be interactive, taking into account the capabilities (e.g., storage capacities and the maximum recording rates) of the clients that receive the program, and using these capabilities as constraints for the determination of the transmission rates.

In its most basic embodiment, the present invention is a modification of the NVOD scheme of DeBey that smoothes out the instantaneous transmission bandwidth

to always be close to the mean transmission bandwidth. This is accomplished by broadcasting all the segments concurrently, starting at the first time interval, but taking n time intervals to broadcast the n-th segment. In other words, if the first segment is transmitted at a transmission rate of T bits per unit time, then the n-th segment is transmitted at a transmission rate of T/n bits per unit time. This is illustrated in Figure 2. Note that all transmissions end at the same times as in Figure 1, but start at different times. Specifically, all the first transmissions of all the segments start at the beginning of the first time interval; and subsequently, as soon as the transmission of any segment ends, that segment is immediately retransmitted.

The simultaneous commencement shown in Figure 2 is illustrative, and applies primarily to the basic embodiment illustrated therein. More generally, because the segments are transmitted repetitively and concurrently throughout the period during which the program is being offered to the viewers, the starting times of the transmission of different segments may be chosen at will.

The most basic embodiment of the present invention relieves the burden on the server, but not necessarily the burden on the clients. For example, a client that tunes in must receive and record all the data being transmitted until such time as it has finished recording a given segment and can cease recording the data for that segment. Thus, this client must be capable of receiving and recording data at a rate equal to the server's aggregate transmission rate. Furthermore, towards the middle of the program, the client's storage medium must have sufficient capacity to store a significant portion of the program even if the client discards the data of a segment once it is displayed. These stringent requirements may be relaxed if the start of the recording by a client of later segments is delayed long enough for earlier segments to be recorded, displayed and possibly even discarded, thus freeing up client resources to accommodate the later segments. Clearly, this implies an increase in the aggregate transmission rate, because a segment must be transmitted in its entirely during the time in which it is being recorded by the client; so that shortening this time without changing segment size mandates an increase in its transmission rate. In fact, the optimum overall system design is a tradeoff between minimizing the aggregate transmission rate and meeting client constraints such as limited recording rate and limited storage capacity. The ability to optimally design the system to optimize certain aspects of performance

while adhering to constraints on others is a salient feature of the current invention. Algorithms for minimizing the aggregate transmission bandwidth subject to these constraints are presented in the Appendix.

More generally, the scope of the present invention includes NVOD methods and systems in which the first segment is transmitted at a higher transmission rate than the subsequent segments. Preferably, transmission rates of successive program segments are such that the latest time, measured from the time of a client's viewing request, at which the client may begin to record a segment such that the recording of the segment is completed before that segment must be displayed, is no earlier for a later program segment than for an earlier program segment. Furthermore, the transmission rates of successive program segments preferably are such that the latest time at which a client must begin to record a segment to ensure that the recording of the segment is completed before the segment must be displayed, *i.e.*, no later than the end of the display of the immediately preceding segment, is no earlier for a later program segment than for an earlier program segment. This display of consecutive segments with no pause in between segments is termed herein "consecutive display" of the segments. It should be noted that this transmission rate of a segment is defined in terms of the overall transmission rate of the segment as a whole. For example, a server may use time division multiplexing to transmit all the segments concurrently on a single channel, by partitioning the segments into subsegments, interleaving the subsegments and transmitting all the subsegments at the same rate. Consider, for example, in the context of the most basic embodiment of the present invention, a case in which 100-second segments are partitioned into 1-second subsegments of B bits each, that are transmitted on a 1 gigabit-per-second channel. The duration of the transmission of each subsegment is $B \times 10^{-9}$ seconds, no matter which segment is the source of the subsegment; but the 100 subsegments of the first segment are transmitted over the course of 100 seconds, for an overall transmission rate of B bits per second, whereas the 100 subsegments of the second segment are transmitted over the course of 200 seconds, for an overall transmission rate of $B/2$ bits per second.

It also is clear that a client may ignore, and not record, transmitted copies of any particular segment or a part thereof, until the transmission of the last such copy that is transmitted in its entirety before the segment must be displayed. To facilitate

this, each transmitted copy of the segment includes metadata that describe various aspects of the segment, including, for example, the segment number, the size of the segment, the transmission rate of the segment, and at least one temporal value related to the time interval between the start of the transmission of this copy of the segment and the start of the transmission of the next copy of the segment. Examples of this temporal value include the length of this time interval itself and an upper bound on this length. If the segments are of unequal length, two temporal values may be associated with each segment: the segment length and the transmission rate assigned to the segment. Similarly, in an embodiment of the present invention in which the segments are partitioned into subsegments, each copy of a subsegment includes metadata that describe various aspects of the subsegment, including, for example, the number of the segment to which the subsegment belongs, the sequence number of the subsegment within its segment, the size of the subsegment, and a temporal value related to the time interval between the start of the transmission of this copy of the subsegment and the start of the transmission of the next copy of the subsegment.

When a segment is not partitioned into subsegments, the preferred embodiments of the current invention nonetheless include metadata that provides sufficient identification so as to permit the recording of the segment's data to begin without waiting for the beginning of the segment. Similarly, when a segment is partitioned into subsegments, the metadata permits the subsegments to be recorded by the client in any order and to subsequently be assembled to form the original segment. Thus, the recording of a given segment by a client may commence essentially as late as one transmission time of the entire segment prior to the earliest time at which that segment may have to be displayed. The time to transmit the entire segment is, in turn, essentially equal to the size of the segment divided by the rate at which the segment is transmitted. Similarly, as soon as a segment is displayed, that segment may be deleted from the recording medium to free up the storage space occupied by that segment.

In one preferred embodiment of the present invention, the subsegments of at least one program segment are used to compute a larger number of such subsegments, such that the original segment can be derived from any sufficiently large subset of the subsegments; and all the subsegments are transmitted such that the time between successive transmissions of the same subsegment remains unchanged. This increases

the transmission rate of the segment, but permits the use of error correcting codes to receive the segment successfully even in the event of communication problems.

In one application of the present invention, in order to allow a viewer to tune in to a broadcast of an entire live program after the program has started but before the program has ended, the basic embodiment of the present invention is initiated simultaneously with the live broadcast, with the repeated transmission of each segment initiated after the live transmission of that segment. This application of the present invention extends naturally to the transmission of several consecutive live programs.

In some embodiments of the present invention, only one copy of each segment is stored at the server. In other embodiments of the present invention, multiple copies of the program data are stored at the server, to increase the efficiency of storage access when transmitting the program.

Optionally, the segments are encrypted and/or compressed by the server prior to transmission, and are decrypted and/or decompressed by each client prior to display.

A system of the present invention includes a software module that embodies the algorithm of the present invention, a processor for executing the instructions of the software module, and a server that includes both a data storage area for storing the program segments and a mechanism for transmitting the program segments according to the transmission rates assigned by the algorithm. Preferably, the software module and the processor also are included in the server; although it also is possible, within the scope of the present invention, for the server to obtain a rate allocation table or a transmission schedule from another device that includes the software module and the processor of the present invention and that operates off-line. The system also includes, for each viewer, a client for receiving, recording and displaying the segments, and a distribution network for broadcasting the segments to the clients.

The scope of the present invention also includes a method for pausing and resuming the display of the program by a client, whether the NVOD transmission rates are selected according to the teachings of the present invention or according to prior art schemes such as DeBey's. Essentially, the client stops displaying the program, but continues to record incoming segments as though the program were being displayed at

the point, in the program, where the display was temporarily stopped. To conserve the client's storage space, metadata are transmitted along with the segments to indicate when the segments will be transmitted again, and segments that will be transmitted and recorded again before they need to be displayed, if displaying were to be resumed immediately, are discarded. This pausing facility is particularly useful in conjunction with a live program: a viewer has the option of toggling between the live program and the delayed segments.

Although the present invention is described herein in terms of the transmission of video data, the term "program", as used herein, encompasses all forms of data, including audio data and textual data, that may be transmitted sequentially to one or more clients and which are such that only a subset of the data, starting from the beginning of the data, must be received in order for the data to begin to be useful. For example, the present invention may be used to transmit the text of a book to several clients. Each client must receive only the first page of the book in order for the client's "viewer" to start reading the book.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is herein described, by way of example only, with reference to the accompanying drawings, wherein:

FIG. 1 is a graphical representation of a prior art NVOD scheduling scheme;

FIG. 2 is a graphical representation of the NVOD transmission scheme of the present invention;

FIG. 3 is a schematic diagram of a system of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention is of a NVOD method and system for transmitting a program to a plurality of viewers for on-demand viewing. Specifically, the present invention can be used by the viewers to view the program continuously, without undue burden on system resources.

The principles and operation of NVOD according to the present invention may be better understood with reference to the drawings and the accompanying description.

The Appendix presents an analysis of the present invention, and preferred methods for assigning transmission rates to program segments according to the present invention, for the special case of a program partitioned into segments of equal length. Nevertheless, the scope of the present invention includes the use of segments of unequal length.

The analysis and the methods presented in the Appendix are directed towards three system parameters, the aggregate server transmission bandwidth R_t, the peak client recording rate R_r and the required client storage capacity S_{max}. Specifically, the algorithms presented in the Appendix are for minimizing R_t subject to the constraints of particular values of R_r and S_{max}. Note that the terms "aggregate transmission rate" and "aggregate transmission bandwidth" are used interchangeably herein. The program is partitioned into L sequential segments indexed by an index m. The time at which the client begins recording segment m is denoted by $s(m)$. The time at which a viewer begins to view segment m is denoted by $v(m)$. In particular, $v(1)$ is equal to the permissible delay from the time of a viewer's request to view a program until the viewer's client begins displaying the program. Recording by the client begins immediately upon the viewing request. The rate at which a particular segment m is transmitted by the server is denoted by $r_t(m)$. In order to guarantee that all the data of segment m is present at the client when the segment is displayed, $s(m) < v(m)-(\textit{size of segment } m)/r_t(m)$. As noted above, the principle of the present invention, in its most general form, is to use $r_t(m>1) < r_t(1)$. Also as noted above, it is preferable to assign $r_t(m)$ such that $s(m_2) \geq s(m_1)$ if and only if $m_2 \geq m_1$, i.e., the times at which the recording of successive segments begins form a monotonically non-decreasing sequence. In the basic embodiment, $r_t(m)$ is proportional to $1/m$. The algorithms presented in the Appendix minimize R_t, subject to the constraints, by systematically increasing $s(m)$ of the basic embodiment (thereby increasing $r_t(m)$) to ensure that the amount of storage in the client's recording medium occupied by recorded segments never exceeds S_{max} and/or that the rate at which the client records incoming segments never exceeds R_r. These algorithms also can be used by one skilled in the art to minimize any one of the client storage requirement, or the client's peak recording rate, subject to constraints on R_t and the other parameters.

Preferably, segments that are transmitted concurrently by the server are multiplexed on a common channel by a conventional multiplexing method such as time division multiplexing, frequency division multiplexing or code division multiplexing. As noted above, if time division multiplexing is used (and optionally if a different multiplexing method is used), each segment is partitioned into sequential subsegments. These subsegments need not be transmitted in their sequential order, although it is preferable that the transmission order of the subsegments be a cyclic permutation of the sequential order of the subsegments, and it is most preferable that the transmission order of the subsegments be identical to the sequential order of the subsegments. Also as noted above, each subsegment include metadata that indicates the sequential order of that subsegment within the corresponding segment, so that a client can start receiving a stream of subsegments in the middle of the sequence, and need not wait for the transmission of the sequence to start again from the beginning of the sequence.

Although the descriptions, herein and in the Appendix, of the present invention, all refer to a situation wherein a client does not store any portion of the program prior to the viewing request, the present invention also applies to the case in which the client does store a portion of the program in advance of viewing. Specifically, if the client stores the first t seconds of the program in advance, the present invention can be applied to the remainder of the program, excluding the first t seconds, with the first subsequent segment labeled as segment number one. The viewing time v still is measured from the time at which the viewing request is made, but $v(1)$ is equal to the permissible delay from viewing request until the commencement of the display of the program plus t. The viewing times of later segments then differ from $v(1)$ by the same amount as when no portion of the program is stored at the client prior to the viewing request.

Referring again to the drawings, Figure 3 is a schematic block diagram of a system **10** of the present invention. The high level components of system **10** include a server **12**, a distribution network **30** and several clients **40**.

Server **12** stores the program and transmits the program to distribution network **30** according to the NVOD transmission rate assignments of the present invention. Server **10** includes a processor **24**; one or more storage/memory units

indicated collectively by reference numeral **14**; a set of input/output devices, such as a keyboard, a floppy disk drive, a modem and a video monitor, represented collectively by I/O block **26**; and a mechanism for transmitting the program to clients **40** via distribution network **30**, represented by a transmitter **28**. Memory **14** includes an instruction storage area **16** and a data storage area **22**. Within instruction storage area **16** is a software module **18** including a set of instructions which, when executed by processor **24**, enable processor **24** to transmit the program in accordance with the teachings of the present invention.

Using the appropriate input device **26**, source code of software module **18**, in a suitable high level language, for implementing the transmission control algorithms of the present invention and for actually transmitting segments of the program in accordance with the output of one of the algorithms, is loaded into instruction storage area **16**. Selecting a suitable language for the instructions of software module **18** is easily done by one ordinarily skilled in the art. The language selected should be compatible with the hardware of server **12**, including processor **24**, and with the operating system of server **12**. Examples of suitable languages include but are not limited to compiled languages such as FORTRAN, C and C++. If a compiled language is selected, a suitable compiler is loaded into instruction storage area **16**. Following the instructions of the compiler, processor **24** turns the source code into machine-language instructions, which also are stored in instruction storage area **16** and which constitute a portion of software module **18**. Using the appropriate input device **26**, the program is loaded into data storage area **22**. Following the machine-language instructions of software module **18**, processor **24** partitions the stored program into segments, schedules the transmission of the segments and directs transmitter **28** to transmit the segments according to the schedule.

Distribution network **30** carries the transmitted segments to clients **40**. Distribution network **30** is illustrated in Figure 3 as a satellite network, including at least one satellite **32**, an uplink channel **34** from server **12** to satellite **32** and multiple downlink channels **36** from satellite **32** to clients **40**. Alternatively, distribution network **30** is a cable television distribution network, a network based on telephone lines, a network based on optical fibers, a special-purpose network, or any suitable

local-, metropolitan- or wide-area network. Preferably, distribution network **30** supports the transmission of information packets to multiple recipients.

Each client **40** includes a receiver unit **46** for receiving the transmitted segments (for example, from downlink channel **36**); a recording medium, represented in Figure 3 as a hard disk **44**, for storing received segments until those segments are displayed; a video display unit **42** for displaying the recorded segments consecutively in their sequential order; and a microprocessor-based control unit **48** for controlling the overall operation of client **40**. Receiver unit **46** is chosen to be appropriate to distribution network **30**. For example, if distribution network **30** is a satellite network, as shown, then receiver unit **46** is a satellite transmission receiver, and if distribution network **30** is a cable television distribution network, then receiver unit **46** is a modem. Client **40** receives the transmitted data, selects the data that should be recorded on recording medium **44**, reorders the data to reconstruct the original program sequence, decompresses the data if necessary, and displays the data on video display **42**. Additionally, client **40** is responsive to viewer commands (play, pause, stop, rewind) directed to control unit **48** via a conventional input device (not shown) such as a keyboard or a remote control device. Control unit **48** makes its decisions based on the viewer commands it receives, on the viewing time ν, on the contents of recording medium **44**, and on the scheduling information that is embedded in the received data or that is supplied separately. After displaying a segment, client **40** either discards that segment or continues to save the segment in recording medium **44** in support of rewind. Client **40** may be an autonomous unit such as a suitably configured personal computer or television set. Alternatively, components **44**, **46** and **48** of client **40** may be embodied in a stand-alone unit such as a television set-top box, for use with a television set that serves as video display **42**.

Because the clocks of server **12** and client **40** may operate at slightly different rates, and because server **12** transmits the same data to all clients **40**, client **40** preferably is capable of synchronizing to the server clock.

In addition to the segments and the metadata, the information transmitted by server **12** to clients **40** optionally includes request/billing information that enables a viewer to select a program from a menu of offered programs and that enables the

provider to record and charge for the transaction. This information may also include a "password" that server **12** transmits to a client **40** to enable that client **40** to display a particular program.

Typically, client **40** uses a secondary storage device such as a hard disk as its main recording medium **44**. Client **40** may also have a substantial amount of primary (random access) memory. Preferably, client **40** uses the available primary memory in order to buffer certain program segments until their playback time, thus obviating the need to store these program segments in main recording medium **44** and read these program segments from main recording medium **44**. This is particularly useful for the earliest segments of the program, because these segments are received and played at a time in which the client recording rate is at its maximum and hard disk bandwidth is stressed. In this preferred embodiment, the main memory that is available for this purpose is used to store the newly received segments whose playback time is the earliest. If received data belongs to an earlier (in the program and thus in terms of playback time) segment than one whose data is already being buffered in the primary memory, the data of the latter is evacuated to main recording medium **44** in favor of the newly received data. In this embodiment, the transmission-rate allocation algorithm is modified to reflect the fact that certain early segments of the program are never recorded in main recording medium **44** or read from main recording medium **44**. This permits the aggregate transmission bandwidth from server **12** to be reduced if that aggregate transmission bandwidth was originally determined by the constraint on recording rate. The peak storage requirement at client **40** is unaffected by the choice of storage medium for early program segments.

As noted above, the scope of the present invention includes the partition of the program into segments of unequal length. Appropriate modifications must be made to the algorithms described in the Appendix to account for segments of unequal length. Each such segment is characterized by a size (in bytes) and the time (in seconds since the start of playing of the program) at which the display of the segment must begin. The transmission rate assigned to a segment is obtained by dividing the size of the segment by the difference in time between the time at which display of the segment begins and the time at which the recording of the segment begins: $r_t(m) = size(m)/(v(m)\text{-}s(m))$. Optionally, the sequence of points in time at which the

algorithm is applied are the (irregularly spaced) points in time at which the displays of the different segments begin.

In its basic form, the present invention assumes that the entire program is available to server **12** at the outset, *e.g.* in a file. Thus, the transmission of all segments can take place in an effectively concurrent manner, and a client **40** can begin recording any segment at any time. There are, however, situations wherein it is desired to permit the viewing in NVOD mode to commence even before the original (live) event has been completed. For example, it is desired to permit the viewing of a one-hour news program to begin at any time after its actual beginning, not only after it is over. The basic NVOD scheme does not work in this case, because the recording of (particularly late) segments by client **40** must begin before the corresponding real event took place.

In order to permit delayed viewing to begin even while the live event is still taking place, the basic scheme is extended as follows:

The live event is transmitted once, as it takes place, as a single unit, at the nominal video rate. The transmission of the first live segment also triggers the beginning of the (infinitely repetitive) NVOD schedule of the basic scheme. Segments that have yet to happen are simply not transmitted. The two constituent transmissions are integrated to form a single data stream. This stream, in addition to the data itself, contains information that permits client **40** to know when a given block of data will be transmitted again, or equivalent information that enables client **40** to decide whether client **40** must record the current instance of a given piece of data. A joining client **40** receives both the remainder of the "live stream" and the basic NVOD transmissions. This client **40** records, from both sources, data that will not be received again before the earliest time at which this client **40** may need such data for viewing. As soon as the live event is over, the "live stream" terminates and system **10** again operates in the basic NVOD mode.

The aggregate transmission rate is initially lower than with the basic NVOD scheme, because there is hardly any data to transmit aside from the "live" stream. As the live event progresses, however, the rate increases and eventually reaches the full NVOD rate plus the live video rate. Once the live event is over, the aggregate transmission rate drops to the NVOD rate. In a most preferred embodiment of the

present invention, the amount of additional transmission bandwidth is reduced by increasing the individual NVOD transmission rates, of program segments that are already available because the corresponding part of the live event has already occurred, during the initial portions of the live event and gradually reducing the NVOD transmission rates to below the original NVOD transmission rate toward the later parts of the live event. The exact details vary between implementations and can be worked out by one ordinarily skilled in the art.

As an important special case of delayed viewing, consider a live TV talk show in which the host and the guests receive and answer telephone calls from viewers. A viewer tunes in to the show, finds the current call intriguing, and wants to view the program from the beginning of the current call, rather than from the beginning of the whole program. To enable this viewer to do this, the program is treated as a collection of consecutive subprograms, with each subprogram corresponding to a different call. Each subprogram is segmented and transmitted separately in NVOD mode as taught herein. Note that, in general, this requires the simultaneous transmission of several subprograms in NVOD mode.

It is well known that communication channels are imperfect due to a variety of reasons. In order to overcome errors, it is common to use error correcting codes. These entail the generation of redundant information that is derived from the original information, such that a sufficiently large subset of the total (redundant plus original) information suffices for the reconstruction of the original information. The aggregate transmission rate is increased in a manner commensurate with the degree of redundancy. For example, if the amount of data transmitted for each segment is increased by a factor of $(k+r)/k$, then the transmission rate must be increased by the same factor. In some preferred embodiments of the present invention, such codes are constructed by dividing each program segment into subsegments of equal sizes, and deriving redundant subsegments from those subsegments. This technique is advantageous over the more common one, wherein redundant information is appended to each subsegment and serves to overcome errors within that segment. Specifically, the advantage of the technique of the present invention is that the technique of the present invention is more resilient than the more common technique to bursts of errors, because the redundant information is transmitted distantly (in time) from the

original information. Thus, a burst of errors destroys small portions of different segments rather than a large portion of a single segment.

Error correcting codes usually entail the computation of redundant information from the original information and the transmission of the original information as well as the redundant information. A different flavor of codes entails computation of information from the original information such that the total number of bits is larger than the original number of bits, and such the original information can be derived from a sufficiently large subset of the full set of bits. In both cases, repetitive transmission of a segment entails the repeated transmission of the total information derived from the original information. So doing, including the special case of no redundant information, requires that client **40** continue to be active even when pausing. As is described in the Appendix, while pausing, client **40** continues to record portions of segments while discarding other portions of those segments. The reason is that, although the fraction of any given segment that must still be recorded before the segment can be viewed does not change with time while client **40** is pausing, the portion (content) of the segment, that will be transmitted between the current time and the time at which the segment will be viewed if playing is resumed at the current time, changes. For example, if a pausing client **40** has recorded the first half of a segment at the time of pausing, this client **40** still needs the second half of that segment. If this client **40** pauses for a time equal to the transmission time of one quarter of that segment and does not record during that time, the missing fraction remains unchanged. However, because transmission has progressed, the fraction of the segment that will be transmitted between resumption of playing and the viewing time of this segment will comprise the fourth quarter of the segment followed by the first quarter of the segment. The third quarter of the segment will thus be missing.

For the reasons just explained, it is also clear that, because the viewing time is frozen while pausing, whereas transmission by the server continues, certain already recorded portions of some segments are certain to be transmitted again prior to the earliest time at which these segments may have to be displayed subsequent to the resumption of viewing. These portions may be discarded by the client. In order to enable the client to decide whether to discard these portions, the client stores (at all

times) the actual (real) time at which any given portion is recorded, as well as the metadata identifying when that portion will next be transmitted.

It is sometimes desirable to cease the recording when pausing. This may be due to a need to use the client computer for other purposes or due to a need to interrupt the communication. In order to permit this, one preferred embodiment of the present invention uses a code that entails the generation of a sequence of subsegments derived from the original segment such that the sequence is much longer than the number of subsegments that are required for reconstruction of the original segment. Furthermore, any sufficiently large subset of the subsegments suffices for this reconstruction. Thus, as long as the transmission time of the entire sequence of subsegments is sufficiently longer than the pausing time, the subsegments recorded by client **40** prior to pausing and the subsegments recorded by client **40** after resuming will all be different, and reconstruction will be possible. An example of such a code is the well-known Reed-Solomon code. Another suitable type of codes is Tornado codes.

In its basic form, system **10** entails transmission of data by server **12**, such that all the data is received by all clients **40**, and each client **40** selects the relevant portions of the data for recording in its memory or disk buffers. This is referred to as "broadcast and select", and is appropriate for situations wherein the transmission medium is a broadcast channel (e.g., satellite and cable television) and wherein the permissible transmission rate on a single channel equals or exceeds the aggregate transmission rate required for providing the NVOD service and wherein each client **40** is capable of receiving data into memory at a rate equal to this aggregate rate. This is not always the case. Nevertheless, the present invention is also applicable to such situations.

One important configuration in which this problem occurs involves the use of telephone lines to reach the client location. These "digital subscriber loops" are inherently point-to-point, with different (physical or logical) lines going to each client **40**, and the permissible data rate over each such line is lower than the aggregate NVOD transmission rate. One preferred embodiment of the method of the present invention for this situation is to execute the selection function at the distant (from client **40**) end of the telephone line. This end of the line is referred to herein as the

"server side" of the line. By so doing, server 40 still transmits at the same rate as before, but the data rate over the private line of any given client 40 is only the recording rate, which varies with time and is initially higher than the video rate, but is substantially lower than the NVOD transmission rate. In order for this scheme to work, client 40 must advise its "agent" at the "server side" of its request to start viewing and of any pausing and resumption. The agent, located on the server side of the private communication line, acts as a filter on behalf of client 40, receiving all transmitted data but refraining from transmitting to client 40 over the private line at least some of the segments that do not need to be recorded by client 40, *e.g.*, segments that have already been viewed by client 40. Indeed, this agent must be able to receive (but not necessarily store on disk) at the full NVOD transmission rate. Another way to think of this is that the filtration that otherwise would occur in client 40, resulting in the recording of only relevant portions of the received data, is carried out at the server-end of the private line, thereby reducing the data rate on the private line from R_t to a lower value, possibly as low as $r_r(v)$.

Another preferred embodiment is one in which the agent executes all the selection, recording and erasure functions of client 40, and sends to client 40, over the private line, the data that client 40 needs for viewing just in time for viewing. Here again, client 40 must advise the agent of its actions, and the agent must also have the recording and storage capabilities normally required of client 40. Client 40 itself provides the user interface and the apparatus for displaying the program that is streamed to it by the agent.

Another important configuration entails the use of cable television or satellite, such that the entire NVOD transmission can be carried over a single channel and the modem of client 40 can listen to all transmissions, but the permissible data rate between the modem and control unit 48 is lower than the channel rate. With current cable modems made by Motorola, for example, the data rate over a single cable channel is 30Mbit/sec, but the connection between the cable modem and the client computer is a 10Mbit/sec Ethernet connection. One preferred embodiment of the present invention for dealing with such a situation entails the use of multiple destination addresses, as follows:

Each program segment is transmitted at the rate determined by the NVOD transmission-rate allocation scheme to at least one of the destination addresses. Each client **40**, depending on the viewing time of that client **40**, accepts data that is destined to a particular subset of the destination addresses. Because certain segments may be transmitted to more than one destination address and are thus transmitted concurrently more than once, this technique increases the aggregate transmission rate. The segments can be grouped in various ways for this purpose, representing a trade-off between transmission rate, the data rate that a client **40** must receive, and the number of destination addresses for which a client **40** must be able to receive data. Because each transmitted data packet is generally received by some subset of clients **40**, this kind of transmission is referred to as multicast.

In another preferred embodiment of the present invention for dealing with this situation, the microcontroller that is part of the cable modem is programmed to act as the server-side agent that was referred to in the example of digital subscriber loops, passing on to client **40** only those segments that client **40** should record.

Yet another important situation occurs when the permissible data rate over a single channel is lower than the aggregate NVOD transmission rate. A preferred embodiment of the present invention in this case entails the transmission of any given segment over at least one channel. At any given time, any given client **40** receives the data transmitted over a subset of the channels. This subset depends on the allocation of segments to channels and on the point in the program that is being viewed at that time. For a given channel data rate, the assignment of segments to channels represents a trade-off between the aggregate transmission rate and the number of channels from which a client **40** must receive data concurrently.

In one preferred embodiment of the present invention, any piece of data is stored in data storage area **22** of server **12** exactly once, and data are read for transmission as required by the transmission rates of the various program segments. This storage scheme minimizes storage requirements in server **12**; but if data storage area **22** is implemented as a hard disk, this storage scheme requires numerous seek operations in order to bring the disk reading head to the location of the data that are to be transmitted. In another embodiment of the present invention, wherein each segment is partitioned into subsegments and the concurrent transmission of the

segments at their respective transmission rates is carried out by appropriately interleaving the transmission of their subsegments over a common channel, the program data are recorded sequentially on disk in the order in which the interleaved subsegments are transmitted over the channel. The recorded sequence is of sufficient length, on the order of the viewing time of the entire program. This substantially increases storage requirements in data storage area **22** of server **12**, but requires no seek operations and thus substantially increases the hard disk's effective data rate.

As noted above, in order for client **40** to decide which data client **40** should record and to correctly reassemble the program, client **40** must know the location within the program of arriving data and whether this data will be received again prior to the earliest possible time at which this client may have to play it for viewing. Again as noted above, in one preferred embodiment of the present invention, each program segment is partitioned into subsegments. Additionally, each transmitted subsegment contains a unique sequence number that identifies the sequence order of that subsegment in the program, as well as the elapsed time from the current transmission of that subsegment until that subsegment will be transmitted again. In another preferred embodiment of the present invention, each subsegment is tagged with its sequence number in the program and also contains that size (in bytes) and transmission rate assigned to the segment to which that subsegment belongs. In yet another embodiment, a subsegment is tagged with its sequence number and server **12** periodically transmits metadata advising clients **40** of the transmission rate assigned to each segment.

When the "long" error-correcting code sequences are used, the time until next transmission of a particular subsegment is irrelevant. Instead, if the degree of redundancy is such that the frequency with which subsegments of a segment are transmitted is greater by a factor of $(k+r)/k$ relative to the frequency in the absence of error correction, the preferred embodiment of the present invention is as follows: each subsegment contains its sequence number in the program and the time until the next transmission of the subsegment (of the same segment) that is $k+r$ later than itself in the code generated for this segment. In the event that the end of the code sequence is reached, counting continues with the first subsegment in the code sequence for this segment. Another embodiment of the present invention entails the transmission of the

sequence number, the segment size (size of $k+r$ subsegments) and the transmission rate allocated to each segment.

Instead of discarding program segments that have been viewed, one preferred embodiment of the present invention retains these segments in recording medium **44** of client **40** in order to permit re-viewing of portions that have been viewed. In one variant of this preferred embodiment of the present invention, segments are only discarded when the storage space devoted to storing the segments in recording medium **44** would otherwise exceed the specified limit (note that the storage space is unused at the beginning and at the end of the display of the program, so there are long periods during which there is extra space). Also in this variant, the order in which segments are discarded is the same as their order in the program. In another variant of this preferred embodiment of the present invention, a minimum required rewind distance is specified and the corresponding amount of storage space in recording medium **44** is reserved for this purpose. The transmission scheme is designed based on an available storage space in recording medium **44** that is equal to the specified available storage space minus the reserved space. The discarding policy is the same as in the first variant.

System **10** optionally includes means of encrypting the transmitted data or a part thereof such that a client **40** must receive authorization from the service provider in order to be able to view a given program. This authorization may be in the form of a password, and may be transmitted through the same channel as the program or provided by other means. According to one embodiment of the present invention, server **12** transmits, with each program, information pertaining to the charge for viewing that program, and the viewer uses a smart card (cash card or credit card) to pay, with the payment occurring at the client site, to enable client **40** (which is trusted by server **12**) to present the program to the viewer.

The embodiments presented above of the NVOD transmission are based on accommodating the least capable client **40**. Thus, all clients **40** can successfully participate. However, there are situations in which accommodating the less capable clients **40** (*e.g.*, clients **40** that have much smaller storage space than the other clients **40**) would result in an excessive transmission rate. The alternative is reduction in video rate (more lossy compression and lower quality), which may be unacceptable to

the capable clients **40**. To alleviate this problem, an extension of the NVOD system and method of the present invention entails the use of progressive encoding of the program segments. According to this well-known method in the art of data compression, a fraction of a compressed segment suffices for its reconstruction, albeit at a reduced quality. In one embodiment of the present invention, every segment of the program is compressed according to a progressive compression scheme and is partitioned into two pieces such that the first piece of any given segment suffices for the reconstruction of that segment, albeit at a lower quality. The first pieces of all segments are then treated as one program, and the second pieces of all segments are treated as a second program. The first pieces are assigned transmission rates so as to satisfy the resource constraints of the "poor" clients **40** while minimizing aggregate transmission bandwidth. Concurrently, the "program" containing the second pieces of every segment is assigned transmission rates so as to satisfy the constraints of the "rich" clients **40**, having first subtracted from those the resources taken up by the first "program". (This idea can be generalized by those skilled in the art to more than two levels of clients **40** and qualities.) The exact allocation of resources and size between the first and second pieces of segments is determined by constraints of the compression scheme and also represents a trade-off between the quality received by the less capable clients **40** and the (higher) quality received by the more capable clients **40**.

The pausing algorithm of the present invention is described schematically as follows in the Appendix:

```
Upon receipt of Pause command: {
        freeze the viewing clock at the current value of v;
        while pausing {
                continue receiving and recording data as if viewing at time v;
                drop the "oldest" bits for each segments as new ones arrive
                        (FIFO);
        }
}
Release the viewing clock and resume normal operation;
```

As noted above, v is the time at which the display of the respective segment commences, relative to when display of the program was requested by the viewer. Although the algorithms in the Appendix, including the pausing algorithm, are defined in terms of segments, it will be clear to those skilled in the art how to implement the pausing algorithm in the context of subsegments as described above. For example, the necessity of continued recording during the pause is discussed above in terms of a segment, half of which is recorded by client **40** at the time of pausing.

While client **40** pauses, client **40** behaves as though frozen in time, so that if client **40** requests display of a program of duration T at time t_0, starts displaying the program at time t_1, pauses at time t_2 (t_2-$t_1 < T$), and resumes display at time t_3, the data stored by that client **40** in recording medium **44** at time t_3, when display resumes, is the same as the data that would be stored at time t_3 by a client that requested display at time $t_0+(t_3-t_2)$, started displaying at time $t_1+(t_3-t_2)$ and did not pause. In particular, when display resumes, display commencement times v of segments yet to be displayed are with reference to $t_0+(t_3-t_2)$ rather than to t_0.

The pause facility of the present invention is particularly useful in connection with the transmission of a live event, as described above. A viewer, upon tuning in to a live event in progress, can decide whether to continue watching the live event in real time or to view the event from its beginning by viewing the NVOD segments in order. At any point in viewing the NVOD segments, the viewer may pause the viewing of the NVOD segments and temporarily view the live event in real time instead, returning to the NVOD segments at will.

Furthermore, several "live" events (in this context, the "live" events may be pre-recorded, as long as they are transmitted sequentially in their entirety, as single units) may be transmitted, one after the other, with each event's NVOD segments being transmitted concurrently with the event, as described above. One illustrative application of this idea is in Internet "radio". Considering an Internet radio station to be a server in this context, and considering the listeners to the station to be "viewers" in this context, with the station playing songs sequentially, each song being a single "program", a "viewer", upon tuning in to the station and hearing his or her favorite song, has the option of "rewinding" the song and playing it from the beginning. Just as, in the transmission of a single program by subsegments, as described above, the

subsegments of several concurrently transmitted segments are interleaved in the transmission channel, so, when segments of several programs are transmitted concurrently, these segments are interleaved in the transmission channel.

The metadata transmitted by server **12** includes, for each segment or for each subsegment, information about when (in terms of clock time, not in terms of display commencement time v) that segment or subsegment will next be transmitted. Client **40** records, for each recorded segment or subsegment, the time at which that segment or subsegment was recorded, and the time at which that segment or subsegment next will be transmitted. This enables client **40** to conserve storage space in medium **44**: as needed, client **40** discards segments or subsegments that will be transmitted and recorded again in time to be displayed at their respective display commencement times v referred to $t_0+(t_3-t_2)$.

While the invention has been described with respect to a limited number of embodiments, it will be appreciated that many variations, modifications and other applications of the invention may be made.

APPENDIX

Tailor-Made transmission plans for efficient Near-Video-On-Demand service

Yitzhak Birk and Ron Mondri

Electrical Engineering Dept., Technion
Haifa 32000, Israel
birk@ee.technion.ac.il, mondri@tx.technion.ac.il

Abstract. Near-Video-On-Demand (NVOD) entails the provision of viewing flexibility that approaches that of VOD to an unlimited number of viewers of a "hot" movie at a fixed cost to the provider. "Near" might refer to viewing commencement within 30 seconds from viewer request as well as instant pausing and resumption. NVOD represents an exciting opportunity to service providers, especially when employing broadcast channels such as the cable television infrastructure or satellites. This paper assumes such a channel and that clients can temporarily store portions of the movie. This is typical of personal computers (and of many television sets and set-top boxes in the near future). We present the Tailor-Made method for designing extremely efficient open-loop transmissions for use by the server. Unlike previous approaches in the same framework, which were based on a mathematical model, Tailor-Made is an algorithmic method. Given the movie parameters, the permissible delay until viewing commences, a client's storage capacity and its recording rate (the same for all clients), Tailor-Made produces an allocation of transmission rates to movie segment such that when the segments are all transmitted again and again concurrently at the respective data rates, the client-resource requirements adhere to the constraints while minimizing server transmission rate. It can also be used as an "oracle" for minimizing any one of required client storage, recording bandwidth and delay until viewing commences if transmission rate and the remaining parameters are specified. The results dominate those of all previous schemes in flexibility as well as in resource consumption. In fact, the ability to tightly tailor resource consumption to availability constraints is sometimes key to the practicality of the service in a given environment.

1. Introduction

Video-On-Demand (VOD) is the online version of traditional video-rental services. As with those, each viewer receives a dedicated "copy" of the movie and can view it in a flexible manner, including the ability to pause and resume, rewind, and possibly even fast-forward. With VOD, the "rental" operation is essentially instantaneous, and viewing can begin within seconds of the decision to view.

The "copy" in VOD is a dedicated video stream. This stream is generated by a video server and sent to the viewer over a communication network. An important advantage of VOD over tape rental is the great flexibility in allocation of resources: the maximum number of concurrent viewers is independent of viewing choices, and is limited only by the server's total streaming capacity. The required bandwidth resources both in the server and in the communication network are proportional to the number of concurrent viewers.

The design of VOD systems has focused on the video servers themselves, with special attention to efficient bandwidth utilization of the storage devices. Related research pertains to data streaming over networks.

Near-Video-On-Demand (NVOD)

There are important situations in which a large number of people wish to view the same content during the same period of time, albeit not simultaneously. One example is viewing emergency-preparedness instruction in the hours or days prior to the arrival of a major storm. Another example is a newly released "hot" movie

that is moreover advertised heavily. Yet another example might be a movie whose viewing is assigned as homework, or even a recorded lecture viewed (individually) in class by the students.

Although VOD could be used to address such situations, it is both highly desirable and intuitively possible to do better. The desire stems from the fact that even the total (over all movies) number of concurrent viewers may be much higher than usual, so it would be very costly if not impossible to design the infrastructure (server and communication network) for such peaks. The intuition that something can be done arises from the observation that the many viewers of the "hot" title are viewing the same material concurrently, albeit not simultaneously. The various schemes for doing better than VOD in this situation are dubbed "*Near Video On Demand*". NVOD can be defined as providing an unlimited number of viewers of the same movie similar service flexibility to that of VOD at a reasonable cost to the server and communication network. Ideally, this cost is independent of the number of viewers. For the purpose of viewing movies, we define "near" to mean commencement of viewing within a minute of viewer request as well as the ability to pause and resume at any time. Rewind and fast-forward functions are not a requirement. NVOD presents an exciting business opportunity to service providers: at long last, they will be able to collect unbounded revenues for "hot" titles while keeping costs fixed.

With NVOD, the nature of the communication network affects the extent to which the resource expenditure can be independent of the number of viewers. To this end, we distinguish between two types of networks based on the nature of the underlying physical layer. While NVOD service can be provided over both types, different such schemes are better suited to one type or the other. The types are:

- *Point-to-point networks.* Here, information sent to a given client is "heard" only by it, i.e., it does not reach other clients. Switched data networks with point-to-point links are one example. Another example is the telephone network and the related digital subscriber loop technologies, such as ADSL. The latter provide high communication bandwidth to the home over telephone lines in support of Internet communication as well as services such as VOD. A salient characteristic of such networks is that the communication bandwidth required for sending information to multiple users, even if this information is identical, is proportional to the number of recipients. Efficient routing techniques such as multicast routing can mitigate the deficiency.
- *Broadcast networks.* Here, all transmissions are broadcast, and each client selectively records the information that is intended for it. The two most prominent examples of such networks are coaxial cable distribution networks (cable television) and satellite-based networks. Thus, the bandwidth required for the transmission of the same information to multiple recipients can be independent of the number of recipients. The down side of this approach is that the total available bandwidth is often lower than with point-to-point networks, since there is no spatial multiplexing.

NVOD systems may be divided into two distinct categories: *open-loop* systems and *closed-loop* systems. In open-loop systems there is no feedback from the viewing client to the server, so neither server transmissions nor routing on the network are affected by viewer actions (other than the possible effect on routing due to a viewer joining a multicast group). Open-loop schemes lend themselves most naturally to broadcast-based networks, and are uniquely suited to such networks that have only one-way communication. This is the common case in satellite-based information-dissemination networks. Closed-loop systems permit some feedback that allows the server to adjust to client requests throughout the viewing period. Our focus in this paper is on open-loop schemes, which are best suited for broadcast topologies.

Prior art

Consider an NVOD system that should provide a movie of length L seconds in a manner that permits viewing to commence within D seconds after viewer request. The straightforward approach for providing such a service at a fixed cost is to start a copy of the movie every D seconds. Viewing would simply entail each client's choosing one of the streams. The aggregate transmission rate (in steady state) would be L/D streams. This is independent of the number of viewers but could be prohibitively expensive. For example, a 100-minute movie with a viewing delay of up to 30 seconds woulrequire 200 streams, each at the original video rate. We refer to this scheme as the ***baseline scheme***. Over the past several years, much research has beento NVOD schemes that require far less bandwidth for the same quality of service.

Closed-loop schemes

In [1], a closed loop system based on "batching" was proposed: the server collects viewing requests over a period of time which is bounded from above by the permissible viewing-commencement delay D, but ends earlier if a sufficient number of requests arrive. Once the period ends, a new video stream is launched. With this approach, bandwidth consumption is not independent of viewer requests, but the amortized (over viewers) bandwidth can be bounded from above. The potential advantage of this scheme over the Baseline scheme is that in certain situations, it offers shorter mean viewing-commencement delay at no additional cost.

The scheme of [2] complements (fewer) streams of the Baseline scheme with ad hoc "private" streams. The latter present the movie at a speed that differs from the nominal one by several percents. (Such a difference goes unnoticed by the viewer.) Within a time period D from a viewing request, a new viewer is issued a fast private stream until it catches up with an earlier "Baseline" stream, or a slow private stream until a later "Baseline" stream catches up with it. For example, playing a stream that is faster than the nominal by 5 percent for ten minutes permits the viewer to catch up with a Baseline stream that started 30 seconds earlier. Typically, only the first few minutes of a movie are stored in the non-standard speed format, so the storage overhead is small. On average, this scheme reduces the required bandwidth, but its stochastic behavior makes it difficult to offer guarantees and may intermittently stress the server as well as a broadcast network.

Open-loop schemes

Recently, several interesting *open-loop* schemes have been proposed. These schemes assume that every client has a substantial amount of available storage capacity, which can be used to temporarily store portions of the movie. In such schemes, the server's transmission schedule and the algorithm used by the client to decide whether or not to "record" any given transmitted block jointly ensure that every segment of the movie will be on the user's disk by its viewing time. We next describe several such schemes. For facility of exposition and because that is the only case addressed by some of the schemes, we assume a fixed video rate. We conveniently use "length" to mean both viewing time and amount of data, and the video rate as our unit of data rate.

De Bey [3] suggested to partition a movie into $N=L/D$ blocks of (fixed) length D, where D is the permissible viewing-commencement delay. The server transmits block m, $1\leq m\leq N$, once every m time slots; the duration of a time slot equals D. The client policy is to receive and store all the transmitted blocks that have yet to be viewed. The server and client policies jointly ensure that every block is available to the client at or before its viewing time. The **mean** transmission bandwidth with this scheme for a movie of length L and permissible viewing delay D is

$$\overline{R}_t=\sum_{m=1}^{N}\frac{1}{m}\approx\ln\left(\frac{L}{D}\right).$$

This is a dramatic improvement over the baseline scheme. However, this method imposes heavy requirements on the client hardware. Analysis shows that required storage is almost $0.5L$, and reception bandwidth reaches more than 12 times the video rate for $L/D=100$, whereas in the Baseline scheme it is equal to the video rate and no storage is required. Furthermore, peak transmission bandwidth is much higher than the average transmission bandwidth. Finally, the direct coupling between the length of the movie, permissible viewing delay and the transmission policy prevents any trade-off between the consumption of various resources.

The *Pyramid* scheme [4] entails partitioning the movie into segments such that each segment is α times longer than the preceding one. Every segment is then transmitted at the same data rate. (This, of course, means that a block belonging to a given segment is retransmitted at time intervals that increase geometrically with segment number.) At any given time, the client records data from at most two (consecutive) segments. The value of α is chosen according to the permissible delay and available transmission bandwidth, and to

ensure that the client will always have the upcoming data available in its storage so as to prevent viewing glitches.

The *Permutation Pyramid* scheme [5] is based on the Pyramid. It creates multiple copies of each segment, partitions each copy into fixed size blocks, and interleaves the different copies of the segment such that same-numbered blocks are equispaced in the interleaved stream. The resulting stream is then transmitted at a fixed per-channel data rate. The result is a shorter viewing-commencement delay as well as lower client-storage and reception-bandwidth for the same amount of transmission bandwidth.

The *Harmonic broadcasting* scheme [6] uses segments of exponentially increasing size. This is essentially the block placement scheme of De Bey, viewed from a channel division perspective.

The *Staircase* scheme [7] presents yet another variant of the same general idea. Here, β channels, each with a transmission rate equal to the video rate, are used. The ith channel, $1 \leq i \leq \beta$, is partitioned into 2^i subchannels. Next, the movie is divided into $N=2^{\beta}-1$ equisized segments (this is equal to the total number of subchannels). The segments are then assigned to the channels, one segment per subchannel, in ascending order of segments and channels. Finally, the segments assigned to each channel are interleaved with fine granularity such that the starting points of consecutive segments are staggered by equal distances, and the result is transmitted at the fixed channel rate. Transmissions occur concurrently on all channels. The client monitors the channels and records information that will not arrive again prior to its viewing time. This scheme outperforms the previous ones, except for the transmission bandwidth in the absence of client-storage and recording-rate constraints. In the latter case, the Harmonic scheme requires a lower transmission rate.

A close look at the NVOD problem reveals a multi-dimensional design space. All previously proposed schemes are based on a systematic "model-based" approach, resulting in very limited design flexibility. Moreover, there is no direct relationship between the design parameters in these schemes and the dimensions of the design space, so it is difficult to "navigate" intelligently even within the artificially constrained space.

Like many of the previous schemes, this paper also addresses open-loop NVOD that exploits client storage. It begins by explicitly presenting the design space. Next, unlike the prior art in this framework, we employ a novel algorithmic approach to optimize the transmission scheme. In so doing, we can minimize transmission rate while satisfying constraints on the remaining dimensions. Our algorithms can also be used in iterative procedures aimed at minimizing a different parameter subject to constraints on the remaining ones and a given transmission rate. The remainder of the paper is organized as follows. In Section 2, we analyze the data path and present the design space. In Section 3, we present our algorithms. Section 4 offers a comparison with prior art, and Section 5 offers concluding remarks.

2. The open-loop NVOD design space

The provision of open-loop NVOD server to storage-capable clients entails (preferably) fixed-rate transmission by the server over a distribution network. Every client then selects the portions of the transmitted material that are relevant to it. A client "records" the selected material and plays it to the viewer at the right time. An NVOD solution thus comprises a transmission scheme which is executed by the servc, and a corresponding selection algorithm which is executed by each client.

The server's task entails reading data from disk and transmitting itothe distribution network. As will be seen later, the required transmission rate for a single NVOD movie poses no problem to a high-performance disk drive and any PC. Moreover, in view of the inherently large ratio of clients to NVOD movies, the cost of a server is critical. Therefore, we do not directly associate cost or performance attributes with server resources.

The distribution network is clearly a critical resource, as its bandwidth is an expensive, possibly scarce resource. We note in passing that this bandwidth is often quantized so it may be important to adhere to specific constraints. Thus, transmission bandwidth R_t is an important dimension of the design space.

The demands placed on the user-premise equipment ("client") are of utmost importance. If it is dedicated to the NVOD service, cost-reduction is critical due to the large quantities. If it is embodied in existing

equipment, e.g., a personal computer, then the resource requirements of the NVOD service will determine the fraction of viewers to which the service can be offered.

Aside from decompression, which depends solely on the movie parameters, the main task performed by the client is recording selected incoming data and subsequently reading it out. If the temporary storage is provided by a disk drive, as is likely to be the case for some time, then disk bandwidth and the required storage capacity are the critical client resources. If the storage medium is semiconductor memory, then bandwidth is negligible and only capacity matters. So, we define the peak client recording rate, R_r, and the peak client storage requirement, S_{max}, as two more dimensions. It should be pointed out that once actual viewing begins, the client records data to disk while reading back data for playback, the latter occurring at an essentially fixed rate. Thus, the maximum permissible recording rate is equal to the disk's effective rate minus the video rate.

Even if a disk drive is used as the client's primary medium for temporary storage, it is possible to use the client's memory in several ways in order to mitigate the recording-bandwidth requirements. One could therefore include the available amount of client memory as another design parameter. This use of memory, however, is closely related to a variety of implementation issues and is not addressed in any depth by the previously proposed schemes. In order to facilitate comparison and to focus on the main contributions of this paper, we do not discuss this use of memory in any detail.

3. The Tailor-Made NVOD scheme

Let us divide a given movie into L (small) segments, and use m (for "movie") to denote the segment number. Next, let us refer to the client's time axis, whose origin is at the time that the client begins to receive and record the movie. We denote time on this axis by v (for "viewer"), and use $v(m)$ to denote the time (on the client's time axis) at which the viewer begins to view segment m. For facility of exposition, the movie segments will all be of equal sizes and duration, thus implicitly assuming a fixed video rate. Also, the duration of the viewing of a single segment will be taken as the unit of time, the size of a segment will serve as the unit of storage, and the movie's data rate will be used as the unit of bandwidth. Thus, the viewer views movie segment m at time $v(m)=D+m$, where D denotes the permissible viewing-commencement delay. We initially assume that the viewer does not pause, but this will be relaxed. Finally, we note that the Tailor-Made algorithms that will be presented shortly can and have been adapted to the case of variable video rate and segment sizes.

Client action. The client's task, as in all NVOD schemes, is to decide which arriving information it should record and which should be discarded. In our case, the client is sufficiently aware of the transmission plan or schedule, and records arriving information if and only if it is not guranteed to be transmitted again prior to the earliest time at which it may be needed for viewing.

Server action. The Tailor-Made scheme assigns a transmission rate $r_t(m)$ to the mth segment, and all segments are transmitted concurrently and repetitively by the server without any synchronization among their starting times. (In practice, blocks belonging to the various segments can be time-interleaved on one or several channels.) Thus, the server transmits at a fixed data rate $R_t=\sum r_t(m)$, regardless of whether the video rate of the movie is fixed or variable. The algorithms that will be described shortly produce the values of $r_t(m)$ that minimize R_t while satisfying various client-related requirements and constraints.

Lemma 1: In any feasible open-loop NVOD transmission scheme, $r_t(m)>(1/v(m))$.

Proof: By contradiction. The amount of segment-m data that is transmitted during $v(m)$ seconds is $r_t(m)\cdot v(m)$. Thus, if $r_t(m)<(1/v(m))$, only part of the segment will be recorded in time for its viewing. □

Proposition 2: For any open-loop storage-assisted VOD system, the minimum aggregate transmission bandwidth R_t (expressed in units of the movie's data rate) required to satisfy a viewing-commencement delay D is

$$R_t^{\min} = \ln\left(1+\frac{L}{D}\right).$$

Proof: Follows from Lemma 1 by choosing the minimum values for $r_t(m)$, substituting $v(m)=D+m$, and approximating the summation over m with integration. □

With $R_t=R_t^{min}$, a client must record all incoming data belonging to segments that have yet to be viewed. Unfortunately, this results in a very high recording rate at the beginning, and the peak client-storage requirement is a large fraction of the movie. In order to reduce peak client recording rate and/or the required amount of client storage space, one must increase server transmission rate. Our algorithms will start out from the minimum-rate solution and will strive to minimize the additional transmission bandwidth required in order to satisfy constraints on the peak client recording rate R_t and its peak storage requirement S_{max}. Before presenting the algorithms, we next derive several important relationships.

If a movie segment is allocated a higher transmission rate than the minimum, the client may postpone the time at which it begins to record this segment. With the unconstrained minimum-bandwidth solution, a client's recording rate decreases monotonically with viewing time, as segments that have been viewed are no longer being recorded. Therefore, postponing the commencement of recording certain segments can reduce the peak recording rate. Also, this monotonicity suggests that it is best for the client to start recording any given segment as late as possible and to record it continuously until its viewing time.

Let us now turn to the client-storage perspective. Regardless of when the recording of a segment ends, the storage space that it occupies can only be released once it is viewed. This release time is thus obviously independent of transmission-rate assignments. Thus, deferring the recording of any given segment is favored by the desire to reduce client-storage requirements. For these two reasons as well as the desire to minimize aggregate transmission rate, the recording of a segment should best terminate just before it has to be viewed even when its transmission rate is increased in order to reduce peak client storage and/or recording-rate requirements. Thus, increasing a segment's transmission rate translates directly into a postponement of the commencement of its recording by the client.

Denoting the (viewing) time at which a client begins to record movie segment m by $s(m)$, it follows that

$$r_t(m)=\frac{1}{v(m)-s(m)}=\frac{1}{D+m-s(m)}.$$

The recording rate at viewing time v is given by

$$r_r(v)=\sum_{\substack{m:s(m)<v\\ v(m)>v}} r_t(m)=\sum_{\substack{m:s(m)<v\\ v(m)>v}}\frac{1}{v(m)-s(m)}=\sum_{\substack{m:s(m)<v\\ m>v-D}}\frac{1}{D+m-s(m)}.$$

The amount of required client storage at viewing time v is simply the amount recorded until that time minus the amount of data belonging to movie segments that have already been played. Stated differently, it is the amount recorded until v from movie segments that have yet to be viewed. Therefore,

$$S(v)=\sum_{\substack{m:s(m)<v\\ v(m)>v}} r_t(m)\cdot(v-s(m))=\sum_{\substack{m:s(m)<v\\ m>v-D}}\frac{v-s(m)}{D+m-s(m)}.$$

Proposition 3: Consider two movie segments, m_1 and m_2, with $m_1<m_2$, whose transmission rates and viewing times are such that the their recording-commencement times are equal, $s(m_1)=s(m_2)$. Next, consider increasing the transmission rate of one of them by a small amount δ. Then, the storage requirement $S(v)$ at any given viewing time is either unchanged or reduced; moreover, at any viewing time v, the reduction in $S(v)$ resulting from the increase in $r_t(m_1)$ is no greater the reduction resulting from the same increase in $r_t(m_2)$.

Proof: Recall that the size of a segment is not altered by changing its transmission rate. Next, note that $S(v)$ is affected by an increase in $r_t(m)$ only if $s(m)<v<v(m)$. Because $s(m_1)=s(m_2)$ prior to the change and $v(m_2)>v(m_1)$, it follows that if $S(v)$ is reduced by the increase in $s(m_1)$, it must also be reduced by an increase in $s(m_2)$. Finally, consider a viewing time v at which both changes reduce $S(v)$. The additional amount of data from segment m_i that would have to be recorded after time v as a result of the increase by δ of m_i's transmission rate is $\delta(v(m_i)-v)$, as this is the product of the additional transmission rate and the duration (after v) during which it would apply. Because m_i's size is unchanged, this is equal to the reduction in the amount of data of segment m_i that is recorded before time v, and which therefore occupies storage space at time v. Because $v(m_2)-v>v(m_1)-v$, it follows that the increase in m_2's transmission rate would be more effective in reducing the storage requirement at time v.

□

From the above, one can deduce that any transmission-rate assignment that minimizes aggregate transmission rate subject to constraints on peak client storage requirement would be such that $s(m)$ for a later segment is no earlier than for an earlier one.

When considering a recording rate constraint, recall that the unconstrained minimization of aggregate transmission rate results in $s(m)=0$ for all segments. This, in turn, results in a monotonically decreasing recording rate as the viewing progresses. Deferring the commencement of recording of later segments would reduce the recording rate at early viewing times, which is critical, and increase them at late viewing times, which is generally not a problem.

Based on the foregoing proofs and insights, we have come up with a novel approach for assigning transmission rates to segments, whereby we express the assignments in terms of the resulting recording-commencement times $s(m)$, and need to consider only those cases in which $s(m)$ is monotonically non-decreasing in m. This greatly reduces the complexity of the assignment-optimization algorithms.

In the assignment algorithms that will be presented shortly, we employ several procedures whose details are obvious and are thus omitted.

Update() recalculates all necessary values using the above formulas. Specifically, it calculates $r_r(v)$ and $S(v)$ for all viewing times v between zero and $D+L$, where L is used to denote both the number of segments in the movie and the duration of its viewing. This is done for convenience, and is true for example when the duration of the viewing of each segment is one second. The generalization to other situations is trivial and is omitted for facility of exposition. *Update()* also calculates the aggregate transmission rate $R_t=\sum r_t(m)$. The efficiency of execution of this function is important to the run-time of the algorithm but has no effect on the results. It can be computed incrementally as known to people skilled in the art of numerical computations, which reduces computation time.

Test(). This procedure checks whether the maximum client recording-rate constraint R_r and/or the maximum client storage constraint S_{max} are satisfied at <u>all</u> viewing times. It returns "OK" or "BAD" with the obvious meanings. This procedure is modified slightly and in an obvious manner for each of the rate-assignment algorithms presented below.

MakeBest() saves the current transmission-rate assignments and aggregate transmission rate as the best thus far.

We are now ready to present the transmission-rate assignment algorithms. Based on the relationships just established, we choose to express the transmission-bandwidth allocation to a given segment m, $r_t(m)$, in terms of the viewing time at which a client would have to start recording this segment, $s(m)$. For facility of exposition, we ignore the fine details of indexing, e.g. beginning at $v=0$ or $v=1$.

Consider a movie divided into L segments, and let *Best* denote a data structure that contains the transmission rate for each segment as well as the aggregate transmission rate for the current best known assignment. When the algorithms terminate, this contains the optimal assignment.

Algorithm 1. **Constrained client storage.**

```
1.  Best.Rt=L;        // initialization to a very high value.
2.  for (s(1)=0; s(1)<v(1); s(1)++){ // iterating over possible values of s(1).
3.          for (s(2)=s(1); s(2)<v(2); s(2)++){ // iterating over possible values of s(2) greater than or equal s(1)
4.  . . .
5.                  for(s(m)=s(m-1); s(m)<v(m); s(m)++)
6.  . . .
7.                          for(s(L)=s(L-1);s(L)<v(L); s(L)++){
8.                                  Update();
9.                          if (Test(Smax) =="OK" ){
10.                                         if(Rt<Rt(Best)) MakeBest();
11.                                 }
12.                         }
13. }
14. Print Best;
```

Remark. Since the algorithm can reach the solution of $s(m)=v(m)-1$ for all m, which requires no storage, successful completion is guaranteed. Therefore, we do not check for error conditions.

Algorithm 2. **Constrained client recording rate.**

Same as Algorithm 1, except:

```
9.                          if (Test(Rr))=="OK"{
```

Algorithm 3. **Constrained client recording rate and storage.**

Same as Algorithm 1, except:

```
9.                          if (Test(Smax, AND Rr))=="OK"{
```

Support for pausing

Pausing and instantaneous resumption can be supported while satisfying the constraints and without any server involvement or increase in transmission rate. In the algorithm, it is assumed that every data block in client storage and every arriving block contains metadata that specifies the time until its next arrival. This can be done easily in several ways. For brevity's sake, we only sketch the client algorithm and offer a correctness proof.

Algorithm 4. **Pause** ***(executed by the client!)***

```
1.  Upon receipt of Pause command: {
2.          freeze the viewing clock at the current value of v;
3.          while pausing {
4.                  continue receiving and recording data as if viewing at time v;
5.                  drop the "oldest" bits for each segments as new ones arrive (FIFO);
6.          }
7.  }
8.  release the vieclock and resume normal operation;
```

Remark. In practice, block-level interleaving may be used to transmit the various segments over a single channel. In such a case, a block will be recorded if and only if its next transmission is later than the time at which it would be needed for viewing if viewing were resumed immediately. Similarly, any given stored

block is dropped if its next transmission will definitely occur before the earliest time at which it may be required for viewing.

Theorem 4: Algorithm 4 is correct. Moreover, the ability to pause at any time and resume viewing instantaneously requires no additional resources.

Proof: Consider a situation wherein viewing is paused at time t_l with corresponding algorithm viewing time v_l, and resumed δ time units later. Next, consider the pausing viewer V1 and a second viewer V2 that begins viewing the movie δ time units after V1. Viewer V2 does not pause. Let us examine the storage contents of the two viewers at time $t_l+\delta$.

V1 received ("heard") all the blocks that V2 received since its viewing began, and discarded only those blocks that were guaranteed to be received again before their earliest possible viewing times. (At any given time, Algorithm 4 assumes that viewing may be resumed immediately.) Therefore, the storage content of V1 is a superset of V2's content at all times.

The non-pausing viewing algorithm is correct. Once viewing is resumed, V1 and V2 will continue receiving identical data and both will be executing identical algorithms with identical viewing times (starting at $v=v_l$ on the algorithm's viewing-time scale. This, combined with the previous observation, guarantees correctness of the Algorithm 4.

Next, suppose that at time $t_l+\delta$, the storage content of V1 a proper superset of V2's storage content, and consider data from segment $m>v_l+D$ that is present only in V1's storage. If this data is not transmitted again prior to its viewing time, V2 will have a problem, which contradicts the correctness of the uninterrupted-viewing algorithms. If it is transmitted again, Algorithm 5 (for V1) would have discarded it by time $t_l+\delta$, which contradicts the assumption. Therefore, the storage contents of V1 and V2 at time $t_l+\delta$ are identical. In conclusion, the storage contents of a client that paused at viewing time v and has been pausing for a duration δ is identical to that of a client that began viewing δ time units later, does not pause, and is at viewing time v. This guarantees that the peak storage requirement for a pausing client is equal to that of a non-pausing client.

The proof for recording rate follows directly by recalling that the recording rate is the derivative with respect to time of the storage content. (In fact, the required disk data rate during pause is even lower because no data is read from disk and discarding merely entails manipulation of metadata.) □

4. Results and comparison with prior art

The purpose of this section is twofold: to compare the new Tailor-Made scheme with previously proposed open-loop schemes, and to provide several representative designs.

The comparison with other schemes is complicated by the fact that their results usually do not explicitly refer to all the dimensions of the design space (aggregate transmission rate R_t, viewing-commencement latency D, peak client recording rate R_r, and peak client storage consumption S_{max}). Also, they cannot generate transmission schemes to specification. Therefore, we were forced to reconstruct several design points for each scheme and to then tailor our design to match these points in all but one dimension. The plots presented in papers describing the previously-proposed schemes depict the value of one parameter, e.g., R_t, versus that of another such as D. These plots are somewhat misleading, however, because the remaining parameters are not held constant. All this makes the presentation of comparative plots either meaningless or prohibitively effort-consuming. Instead, we resort to tables with a small number of representative results. The Pyramid scheme [4] is dominated by the Permutation Pyramid [5]. The Harmonic scheme [6] performs well in terms of R_t in the unconstrained case, but even then it is inferior even in this regard to the Tailor-Made scheme. This is due to the coupling between segment length and viewing-commencement delay D. At its only design point for any given combination of L and D, the Harmonic scheme requires a large amount of client storage, rendering it impractical in many situations. In view of this, Talor-Made will only be compared with the Permutation Pyramid and Staircase schemes.

Remark. In order to conform to the format of the results for the other schemes, we cite the total disk rate R_d rather than its recording rate R_r. For Tailor-Made, we use $R_d = R_r + 1$, reflecting the fact that we did not exploit the extra bandwidth that is available for recording while $v<D$.

Tables 1 and 2 present transmission-rate comparisons between the Tailor-Made scheme and the Permutation Pyramid and Staircase schemes, respectively, for equal values of D, R_r and S_{max}. Movie length is 120 minutes, and the comparison is carried out for three values of D: 10, 30 and 60 seconds. S_{max} is expressed in percents of the movie, and both R_t and R_r are expressed in units of the movie's video rate R_v. The advantage of the Tailor-Made scheme is clearly evident. However, a hidden advantage which is often of equal or greater importance is the ability of the Tailor-Made scheme to produce a bandwidth allocation that matches the constraints. The reader should appreciate that carrying out a similar comparison between two of the other schemes would be extremely tedious if not impossible due to the absence of this feature. A similar problem would occur in practice whenever an NVOD service has to be "squeezed" into a set of constraints.

	D[sec]	S_{max} [%]	$R_d[R_v]$	$R_t[R_v]$
Perm. Pyr.	60	23.3	3.65	18.56
Tailor-Made	60	23.3	3.65	5.29
Perm. Pyr.	30	24.3	3.89	20.21
Tailor-Made	30	24.3	3.89	5.93
Perm. Pyr.	10	24.5	4.25	22.74
Tailor-Made	10	24.5	4.25	7.05

Table 1. Transmission-rate comparison: Permutation Pyramid Vs. Tailor-Made. $L=120$ *min.*

	D[sec]	S_{max} [%]	$R_d[R_v]$	$R_t[R_v]$
Staircase	60	24.8	2.63	7
Tailor-Made	60	24.8	2.63	6.14
Staircase	30	24.9	2.75	8
Tailor-Made	30	24.9	2.75	6.83
Staircase	10	25.0	2.83	10
Tailor-Made	10	25.0	2.83	8.18

Table 2. Transmission-rate comparison: Staircase Vs. Tailor-Made. $L=120$ *min.*

In Tables 3 and 4, we focus on a single design point of the Permutation Pyramid and Staircase schemes, respectively. We use Tailor-Made to tailor a design point that is constrained in all but one dimension to the same values as the point of the referenced scheme, and compare the values in the remaining dimension. It can readily be seen that Tailor-Made dominates the referenced schemes. (In the third row of Table 3, $D<1$ sec was deemed impractical, hence the lower value of R_t.)

	D[sec]	S_{max} [%]	$R_d[R_v]$	$R_t[R_v]$
Perm. Pyr.	30	24.3	3.89	20.2
Tailor-Made	30	24.3	3.89	5.9
Tailor-Made	1	24.3	3.89	10.2
Tailor-Made	30	2.7	3.8	20.2
Tailor-Made	30	24.3	2.09	20.2

Table 3. Tailor-Made Vs. Permutation Pyramid. $L=120$ *min.*

	D [sec]	S_{max} [%]	R_d [R_v]	R_t [R_v]
Staircase	30	24.9	2.75	8
Tailor-Made	30	24.9	2.75	6.8
Tailor-Made	14	24.9	2.75	7.9
Tailor-Made	30	12.5	2.75	7.9
Tailor-Made	30	24.9	2.48	7.9

Table 4. Tailor-Made Vs. Staircase. *L=120 min.*

Next, we demonstrate the capabilities and performance of the Tailor-Made scheme by showing several possible configurations for two video rates: 1.2Mb/s (MPEG-1 streams) in Table 5, and 4.8Mb/s (PAL broadcast quality MPEG-2 streams) in Table 6. In choosing the configurations, we targeted two communication fabrics: 10Mb/s Ethernet and a 25-30Mb/s cable channel.

L [min]	D[sec]	S_{max}[MB]	R_d[KB/s]	R_t [Mb/s]
30	10	32	766	8.43
30	30	32	596	7.13
120	10	265	412	10
120	30	162	412	9
120	30	64	710	13
120	30	32	596	22.35
120	30	12	650	27

Table 5. Tailor-Made design points for R_v=1.2Mbit/s (MPEG-1).

L [min]	D[sec]	S_{max} [MB]	R_d [KB/s]	R_t [Mb/s]
120	30	1,600	3,900	26.4
120	60	,600	3,480	23.1
1	60	1,000	2,190	25.4

Table 6. Tailor-Made design points for R_v= 4.8Mbit/s (PAL broadcast-quality MPEG-2).

Referring to the bottom row of Table 5, for example, we see that even with only 12MB of client storage space, one can use a single 6MHz television channel (used with a cable modem) to offer a 120-minute MPEG-1 movie in NVOD mode with $D=30s$. This does not require a disk drive! Referring to the 5th row, two such movies can be offered concurrently if the client has 64MB of available memory, which will be very reasonable in the near future. Referring to the 4th row, 162MB of client storage would permit the provision of three such movies on a cable channel or one over 10Mb/s Ethernet. Tiny magnetic disk drives weighing only 20gr with double this capacity are becoming available. With 1.6GB of client storage, a single MPEG-2 movie with quality similar to broadcast-quality PAL can be offered over a cable channel. Recently, "digital TV" sets with multi-GB disk drives have been announced, so this configuration is also likely to become viable in the near future. Of course, the use of a modem PC as the platform for the client is likely to provide the required resources free of charge. We note in conclusion that the foregoing discussion also illustrated the importance of the ability to design the NVOD system to specifications.

Remark. The results in every row of the tables can be scaled in order to obtain numerous design points. To do so, R_b, R_r, R_v, and S_{max} must be changed by the same multiplicative factor while keeping D and L unchanged.

Finally, it should be noted that the results presented in this section for the Tailor-Made scheme are conservative. No use was made of RAM buffers to mitigate the disk-rate requirement, and we even did not take advantage of the fact that as long as $v<D$, no data is read from disk so its entire

bandwidth can be devoted to recording. Therefore. the advantage of Tailor-Made over the other schemes is actually greater than suggested by the results..

5. Conclusions

This paper presented the Tailor-Made scheme for designing optimal open-loop NVOD systems that utilize client storage. Unlike all previous work, which followed an approach of a structured bandwidth allocation with some parameterization. Tailor-Made is algorithmic. It directly addresses constraints on resource utilization. and minimizes transmission bandwidth subject to those. Of course. one can use the algorithms as an oracle in an iterative search aimed at minimizing any one of storage consumption. recording rate and viewing-commencement delay subject to constraints on transmission rate and the other resources. In the paper. we presented algorithms for the case of a fixed video rate. However, the algorithms have been adapted and work equally well with variable data rates. It is important to observe that, regardless of the behavior of the video rate. the server always transmits at a fixed data rate.

The Tailor-Made transmission schemes are much more efficient than the prior art. In producing the Tailor-Made results for comparison with prior art, we refrained from any special optimizations such as taking into account the fact that the disk does not need to supply data until actual viewing begins. The actual advantage of our scheme is thus even greater than suggested by the comparison. Also. its ability to tailor the transmission schedule to the exact constraints makes the Tailor-Made approach particularly attractive due to quantization problems associated with many resources. For example. there is little meaning to being able to fit one and a half movies into a given cable-TV channel.

References

[1] A. Dan. P. Shahabuddin. D. Sitaran and D. Towsley. "Channel allocation under batching and VCR control in Movie-On-Demand servers". Technical report, IBM Research. 1994.

[2] L. Golubchik. L. C.-S. Lui. and R. R. Muntz, "Reducing I/O Demand in Video-On-Demand Storage Servers". Proceedings of ACM SIGMETRICS, pp. 25-36. 1995.

[3] H. C. De Bey. "Program transmission optimisation". United States Patent Number 5.421.031. Mar 1995.

[4] S. Viswanathan and T. Imielinski, "Metarea video-on-demand service using pyramid broadcasting", Multimedia Systems. vol. 4, pp. 197-208, 1996.

[5] C.C. Aggarwal. J.L. Wolf. and P.S. Yu. "A permutation-based pyramid broadcasting scheme for video-on-demand systems". IEEE Proceedings of the International Conference on Multimedia Computing and Systems, pp. 118-126. Jun 1996.

[6] L-S. Juhn and L-M. Tseng. "Harmonic broadcasting for video-on-demand service". IEEE Trans. Broadcasting. vol. 43, no. 3, pp. 268-271, Sep 1997.

[7] L-S. Juhn and L-M. Tseng. "Staircase data broadcasting and receiving scheme for hot video service". IEEE Trans. Consumer Electronics. vol. 43, no. 4, pp. 1110-1117, Nov 1997

WHAT IS CLAIMED IS:

1. In a system wherein a server transmits a program having a certain duration, the program being received by at least one client, a method for planning the transmission of the program, comprising the steps of:

(a) partitioning the program into a plurality of sequential segments; and

(b) selecting a transmission rate for each said segment, said transmission rate that is selected for a first said segment being faster than said transmission rate that is selected for any other said segment.

2. The method of claim 1, wherein said partitioning of the program is effected by the server.

3. The method of claim 1, wherein said selecting of said transmission rates is effected by the server.

4. The method of claim 1, wherein said segments are of equal length.

5. The method of claim 1, further comprising the step of:

(c) transmitting said segments, by the server, each said segment being transmitted at said transmission rate of said each segment.

6. The method of claim 5, further comprising the step of:

(d) recording said segments, by the at least one client;

and wherein, for each said segment subsequent to said first segment in the program, said transmission rate of said each segment is such that the at least one client must begin to record said each segment at least as late as when the at least one client must begin to record said segments that precede said each segment in the program.

7. The method of claim 5, further comprising the steps of:

(d) recording said segments, by the at least one client; and

(e) displaying said segments, by the at least one client;

and wherein, for each said segment subsequent to said first segment in the program, said transmission rate of said each segment is such that the at least one client must begin to record said each segment at least as late as when the at least one client must begin to record said segments that precede said each segment in the program to ensure that said segments can be displayed consecutively by the at least one client in said sequence.

8. The method of claim 5, wherein at least some of said segments are transmitted by the server at least partly concurrently.

9. The method of claim 5, wherein all said segments are transmitted by the server substantially concurrently.

10. The method of claim 5, wherein at least one said segment is transmitted repeatedly by the server during one duration of the program.

11. The method of claim 5, further comprising the steps of:

(d) recording said segments, by the at least one client; and

(e) playing said recorded segments in said sequence, by the at least one client.

12. The method of claim 11, wherein the at least one client records only one copy of each said segment.

13. The method of claim 12, wherein said only one copy of said each segment is a last said copy received in an entirety thereof by the at least one client prior to said playing of said each segment.

14. The method of claim 5, wherein said selecting of said transmission rates is effected in a manner that substantially minimizes a total bandwidth of said transmitting.

15. The method of claim 14, wherein said minimization of said total bandwidth is effected subject to a constraint based on at least one parameter of the at least one client.

16. The method of claim 15, further comprising the steps of:

(d) recording said segments, by the at least one client, in a recording medium having a certain capacity;

(e) playing said recorded segments in said sequence, by the at least one client; and

(f) for at least one said recorded segment, deleting said recorded segment subsequent to said playing thereof, by the at least one client;

and wherein said at least one parameter includes said capacity of said recording medium, said constraint being that, at any one time, said recorded segments occupy at most said capacity of said recording medium.

17. The method of claim 16, wherein each said recorded segment is deleted immediately subsequent to said playing thereof.

18. The method of claim 16, wherein said selecting of said transmission rates is effected by steps including:

(i) initializing said transmission rates to be as low as possible, consistent with receipt of said segments, by the client, soon enough for said segments to be displayed consecutively; and

(ii) increasing at least a portion of said transmission rates to delay said recording of respective segments sufficiently so that, at any one time, said recorded segments occupy at most said capacity of said recording medium.

19. The method of claim 15, further comprising the step of:

(d) recording each said segment, by the at least one client, at at most a certain recording rate;

and wherein said at least one parameter includes said recording rate, said constraint being that a sum of said transmission rates of each said segment that is recorded by the at least one client at any one time is at most said recording rate.

20. The method of claim 19, wherein said selecting of said transmission rate is effected by steps including:

(i) initializing said transmission rates to be as low as possible, consistent with receipt of said segments, by the client, soon enough for said segments to be displayed consecutively; and

(ii) increasing at least a portion of said transmission rates to delay said recording of respective segments sufficiently so that a sum of said transmission rates of each said segment that is recorded by the at least one client at any one time is at most said recording rate.

21. The method of claim 15, further comprising the steps of:

(d) recording each said segment, by the at least one client, at at most a certain recording rate, in a recording medium having a certain capacity;

(e) playing said recorded segments in said sequence, by the at least one client; and

(f) for at least one said recorded segment, deleting said recorded segment subsequent to said playing thereof, by the at least one client;

and wherein said at least one parameter includes said recording rate and said capacity of said recording medium, said constraint being that, at any one time:

(i) said recorded segments occupy at most said capacity of said recording medium; and

(ii) a sum of said transmission rates of each said segment that is recorded by the at least one client at said one time is at most said recording rate.

22. The method of claim 21, wherein each said recorded segment is deleted immediately subsequent to said playing thereof.

23. The method of claim 5, further comprising the step of:

(d) partitioning at least one said segment into a plurality of subsegments; said transmitting of said at least one segment being effected by transmitting each said subsegment at a subsegment transmission rate at least as great as said transmission rate of said at least one segment.

24. The method of claim 23, wherein said partitioning of said at least one segment is effected by the server.

25. The method of claim 23, further comprising the steps of:

(e) deriving at least one redundant subsegment from said plurality of subsegments, said transmitting of said at least one segment being effected by transmitting both said plurality of subsegments and said at least one redundant subsegment; and

(f) increasing said transmission rate of said at least one segment in accordance with a number of said at least one redundant subsegment that are derived.

26. The method of claim 5, further comprising the step of:

(d) transmitting the program as a single unit, by the server.

27. The method of claim 26, wherein said transmitting of each said segment commences subsequent to a time at which a portion of said single unit corresponding to said each segment is transmitted.

28. The method of claim 26, wherein said transmitting of at least one said segment is effected concurrently with said transmitting of the program as a single unit.

29. The method of claim 26, wherein said at least one said segment, that is transmitted concurrently with said transmitting of the program as a single unit, includes said first segment, the method further including the step of:

(e) receiving both said transmitted segments and said transmitted single unit, by one of the at least one client, subsequent to a start of said transmitting of said single unit, so that said one client receives only a portion of said single unit;

and wherein the method further includes a step selected from the group consisting of:

(f) displaying said portion of said single unit, by said one client; and

(g) displaying said segments in said sequence, starting from said first segment, by said one client.

30. The method of claim 5, further comprising the step of:

(d) storing at least one copy of each said segment, by the server, prior to said transmitting.

31. The method of claim 5, further comprising the step of:

(d) transmitting at least one item of metadata, by the server.

32. The method of claim 31, wherein said at least one item of metadata is included in at least one of said segments.

33. The method of claim 32, wherein said at least one item includes a sequence number of said segment wherein said at least one item is included.

34. The method of claim 33, wherein a plurality of copies of said segment, wherein said at least one item is included, are transmitted, and wherein, in at least one said copy of said segment wherein said at least one item is included, said at least one item includes at least one temporal value related to a time interval between a transmission start time of said at least one copy of said segment wherein said at least one item is included and a transmission start time of an immediately succeeding copy of said segment wherein said at least one item is included.

35. The method of claim 33, further comprising the step of:

(e) partitioning said segment, wherein said at least one item is included, into a plurality of subsegments;

each said subsegment including at least one said item of said metadata.

36. The method of claim 35, wherein said partitioning is effected by the server.

37. The method of claim 35, wherein said at least one item includes a sequence number of said subsegment.

38. The method of claim 35, wherein a plurality of copies of said subsegments of said at least one segment are transmitted, and wherein, in at least one said copy of said at least one subsegment, said at least one item includes at least one temporal value related to a time interval between a transmission start time of said at least one copy of said each subsegment of said segment wherein said at least one item is included and a transmission start time of a succeeding copy of said each subsegment of said segment wherein said at least one item is included.

39. The method of claim 38, wherein said succeeding copy of said each subsegment of said at least one segment, wherein said at least one item is included, is an immediately succeeding copy of said each subsegment of said each segment.

40. The method of claim 35, wherein said at least one item includes a size of said each subsegment..

41. The method of claim 35, wherein said at least one item includes said transmission rate of said segment wherein said at least one item is included.

42. The method of claim 35, further comprising the step of:

(f) deriving at least one redundant subsegment from said plurality of subsegments;

and wherein said at least one item includes a total number of said subsegments that are transmitted during said transmitting of said segment wherein said at least one item is included.

43. The method of claim 5, further comprising the step of:

(d) encrypting said segments, by the server, prior to said transmitting thereof.

44. The method of claim 5, further comprising the step of:

(d) compressing each said segment, according to a progressive encoding scheme, by the server, prior to said transmitting thereof.

45. The method of claim 5, further comprising the step of:

(d) deriving, from at least one said segment, a plurality of subsegments such that any subset, of said plurality of segments, that includes a certain number of said segments less than a total number of said segments, suffices to reconstruct said at least one segment;

said plurality of subsegments then being transmitted in place of said at least one segment.

46. The method of claim 45, wherein said formatting is effected according to a code selected from the group consisting of Reed-Solomon codes and Tornado codes.

47. The method of claim 5, further comprising the steps of:

(d) displaying at least a portion of said segments, by one of said at least one client, starting from said first segment; and

(e) pausing said display, by said one client.

48. The method of claim 47, further comprising the step of:

(f) resuming said displaying, by said one client, subsequent to said pausing.

49. The method of claim 1, further comprising the steps of:

(c) transmitting said segments, by the server, to an agent, each said segment being transmitted at said transmission rate of said each segment; and

(d) retransmitting at least some of said segments, by said agent, to the at least one client, each said segment, that is retransmitted by said agent, being retransmitted at a respective retransmission rate that is at most as great as said transmission rate of said each segment.

50. The method of claim 49, wherein, for each said segment that is retransmitted by said agent, said retransmission rate of said each segment is lower than said transmission rate of said each segment.

51. The method of claim 49, further comprising the step of:

(e) displaying said segments, by the at least one client;

said agent retransmitting to the at least one client only said segments that have yet to be displayed by the at least one client.

52. The method of claim 49, further comprising the steps of:

(e) recording said segments, by the at least one client; and

(f) playing said recorded segments in said sequence, by the at least one client.

53. The method of claim 49, further comprising the steps of:

(e) recording said segments, by said agent, prior to said retransmission of said segments; and

(f) playing said recorded and retransmitted segments in said sequence, by the at least one client.

54. A system for transmitting a program to at least one viewer, comprising:

(a) a software module including a plurality of instructions for transmitting the program by:

(i) partitioning the program into a plurality of sequential segments, and

(ii) selecting a transmission rate for each said segment, said transmission rate that is selected for a first said segment being faster than said transmission rate that is selected for any other said segment;

(b) a processor for executing said instructions;

(c) a server for transmitting each said segment at said respective transmission rate; and

(d) for each at least one viewer, a client for receiving said transmitted segments, recording said received segments and playing said recorded segments in said sequence.

55. The system of claim 54, wherein said instructions are for selecting said transmission rates in a manner that substantially minimizes a total bandwidth of said transmitting.

56. The system of claim 54, wherein said instructions are for selecting said transmission rates in a manner that substantially minimizes a total bandwidth of said transmitting subject to a constraint.

57. The system of claim 56, wherein each said client includes:

(i) a recording medium, having a certain capacity, for recording said segments;

said constraint being that, at any one time, said recorded segments occupy at most said capacity of said recording medium.

58. The system of claim 56, wherein each said client includes:

(i) a recording medium, for recording said segments at a certain recording rate;

said constraint being that, at any one time, a sum of said transmission rates of said segments that are recorded by said client at said one time is at most said recording rate.

59. The system of claim 54, further comprising:

(e) a distribution network for broadcasting the program to said at least one client.

60. The system of claim 54, wherein said software module and said processor are included in said server.

61. In a system wherein a server transmits a program that is partitioned into a plurality of segments, the segments being transmitted repeatedly, and wherein a client receives and records the segments and displays the program, the transmitting, receiving and recording of the segments being effected according to a transmission plan, a method for displaying the program intermittently, comprising the steps of:

(a) transmitting, along with the segments, metadata describing the transmission plan, by the server;

(b) pausing the display of the program, by the client;

(c) resuming the display of the program, by the client, subsequent to said pausing; and

(d) during said pausing, continuing to record at least a portion of the segments then received, by the client.

62. The method of claim 61, wherein said pausing is initiated at a certain pausing time, and wherein said recording is continued in accordance with a time of the displaying of the program remaining fixed at said pausing time.

63. The method of claim 62, wherein said metadata include, for each segment, a next transmission time, the method further comprising the step of:

(e) discarding at least one segment that will be transmitted and recorded again prior to respective said display commencement times relative to immediate said resuming.

64. In a system wherein a server transmits a program that is received by at least one client, a method for planning the transmission of the program, comprising the steps of:

(a) partitioning the program into a plurality of sequential segments; and

(b) selecting a transmission rate for each said segment, said selecting being effected in a manner that substantially minimizes a total transmission bandwidth, subject to a constraint based on at least one parameter of the at least one client.

65. The method of claim 64, further comprising the step of:

(c) transmitting said segments, by the server, each said segment being transmitted at said respective transmission rate.

66. The method of claim 65, further comprising the step of:

(d) recording said segments, by the at least one client;

and wherein, for each said segment subsequent to a first said segment, said transmission rate of said each segment is such that the at least one client must begin to record said each segment at least as late as when the at least one client must begin to record said segments that precede said each segment in the program.

67. The method of claim 65, further comprising the steps of:

(d) recording said segments, by the at least one client; and

(e) displaying said segments, by the at least one client;

and wherein, for each said segment subsequent to a first said segment, said transmission rate of said each segment is such that the at least one client must begin to record said each segment at least as late as when the at least one client must begin to record said segments that precede said each segment in the program to ensure that

said segments can be displayed consecutively by the at least one client in said sequence.

68. The method of claim 65, further comprising the steps of:

(d) recording said segments, by the at least one client, in a recording medium having a certain capacity;

(e) playing said recorded segments in said sequence, by the at least one client; and

(f) for at least one said recorded segment, deleting said recorded segment subsequent to said playing thereof, by the at least one client;

and wherein said at least one parameter includes said capacity of said recording medium, said constraint being that, at any one time, said recorded segments occupy at most said capacity of said recording medium.

69. The method of claim 65, further comprising the step of:

(d) recording each said segment, by at least one client, at at most a certain recording rate;

and wherein said at least one parameter includes said recording rate, said constraint being that a sum of said transmission rates of each said segment that is recorded by the at least one client at any one time is at most said recording rate.

70. A system for transmitting a program to at least one viewer, comprising:

(a) a software module including a plurality of instructions for transmitting the program by:

  (i) partitioning the program into a plurality of sequential segments, and

  (ii) selecting a transmission rate for each said segment;

(b) a processor for executing said instructions;

(c) a server for transmitting each said segment repeatedly at said respective transmission rate; and

(d) for each at least one viewer, a client for receiving said transmitted segments, recording said received segments and playing said recorded segments in said sequence;

and wherein said transmission rates are selected in a manner that substantially minimizes a total transmission bandwidth, subject to a constraint based on at least one parameter of said at least one client.

71. The system of claim 70, wherein each said client includes:

(i) a recording medium, having a certain capacity, for recording said segments;

said constraint being that, at any one time, said recorded segments occupy at most said capacity of said recording medium.

72. The system of claim 70, wherein each said client includes:

(i) a recording medium, for recording said segments at a certain recording rate;

said constraint being that, at any one time, a sum of said transmission rates of said segments that are recorded by said client at said one time is at most said recording rate.

73. A method for transmitting a program from a server to at least one client, comprising the steps of:

(a) partitioning the program into a plurality of sequential segments;

(b) selecting a transmission rate for each said segment;

(c) transmitting the program as a single unit, by the server;

(d) transmitting said segments, by the server, each said segment being transmitted at said respective transmission rate, said transmitting of at least a portion of said segments being concurrent with said transmitting of said single unit; and

(e) for one of the at least one client:

(i) receiving both said transmitted segments and said transmitted single unit, subsequent to a start of said transmitting of said

single unit, so that said one client receives only a portion of said single unit, and

(ii) effecting a step selected from the group consisting of:

(A) displaying said portion of said single unit, and

(B) displaying said segments in said sequence, starting from said first segment.

74. A method for transmitting a plurality of programs from a server to at least one client, comprising the steps of:

(a) partitioning each program into a plurality of sequential segments;

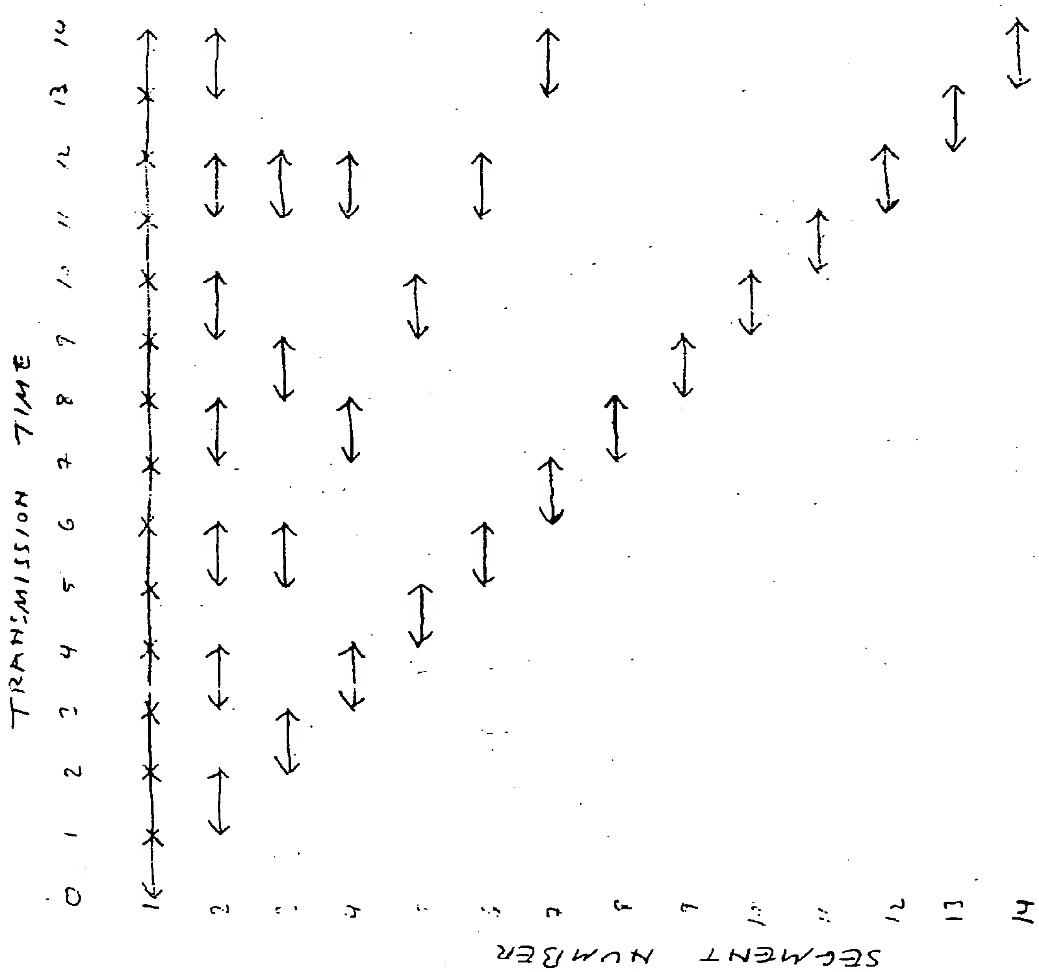
(b) for each program, selecting a transmission rate for each said segment;

(c) transmitting the programs sequentially, by the server, each program being transmitted as a single unit; and

(d) for each program, transmitting said segments of said each program, by the server, each said segment being transmitted at said respective transmission rate, said transmitting of at least a portion of said segments being concurrent with said transmitting of said each program as a single unit.

75. The method of claim 74, wherein, for each program, said transmitting of each said segment commences subsequent to a time at which a portion of said single unit corresponding to said each segment is transmitted.

FIG. 1
(PRIOR ART)



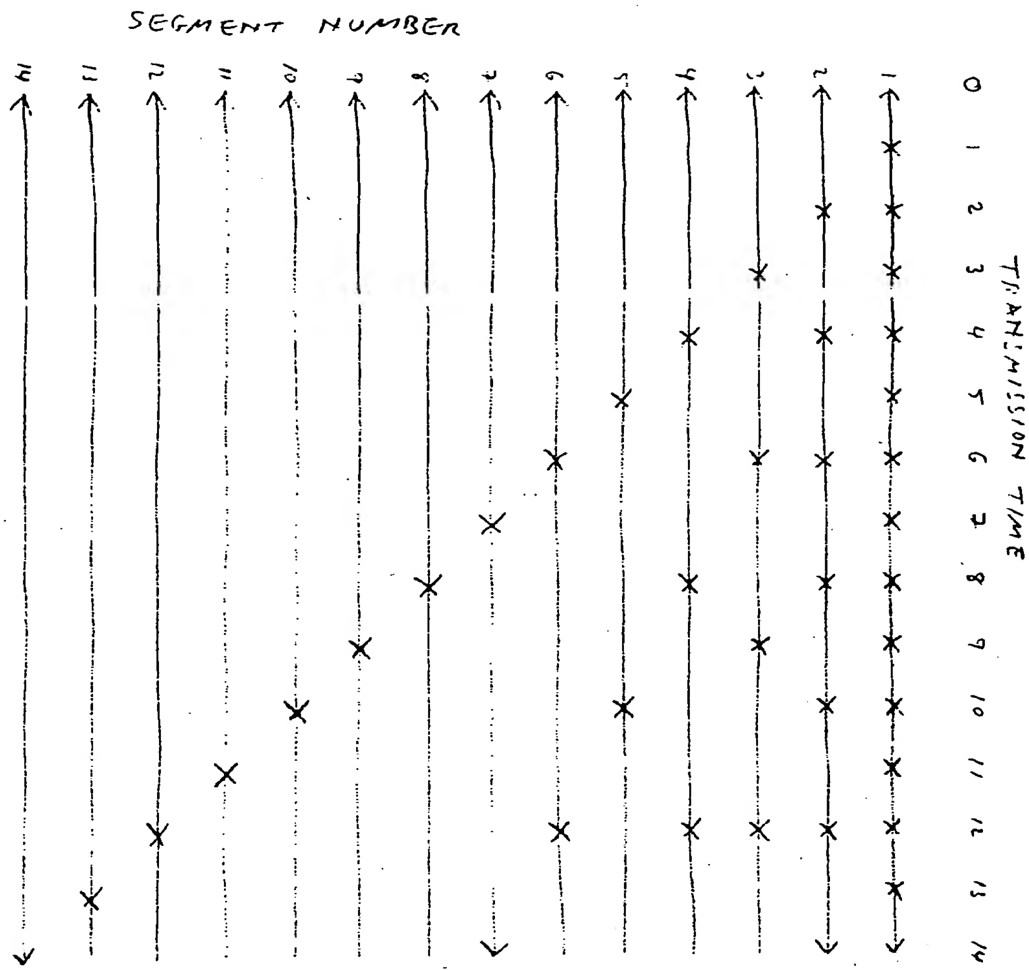

SEGMENT NUMBER
TRANSMISSION TIME
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 2 3 4 5 6 7 8 9 10 11 12 13 14

FIG. 2

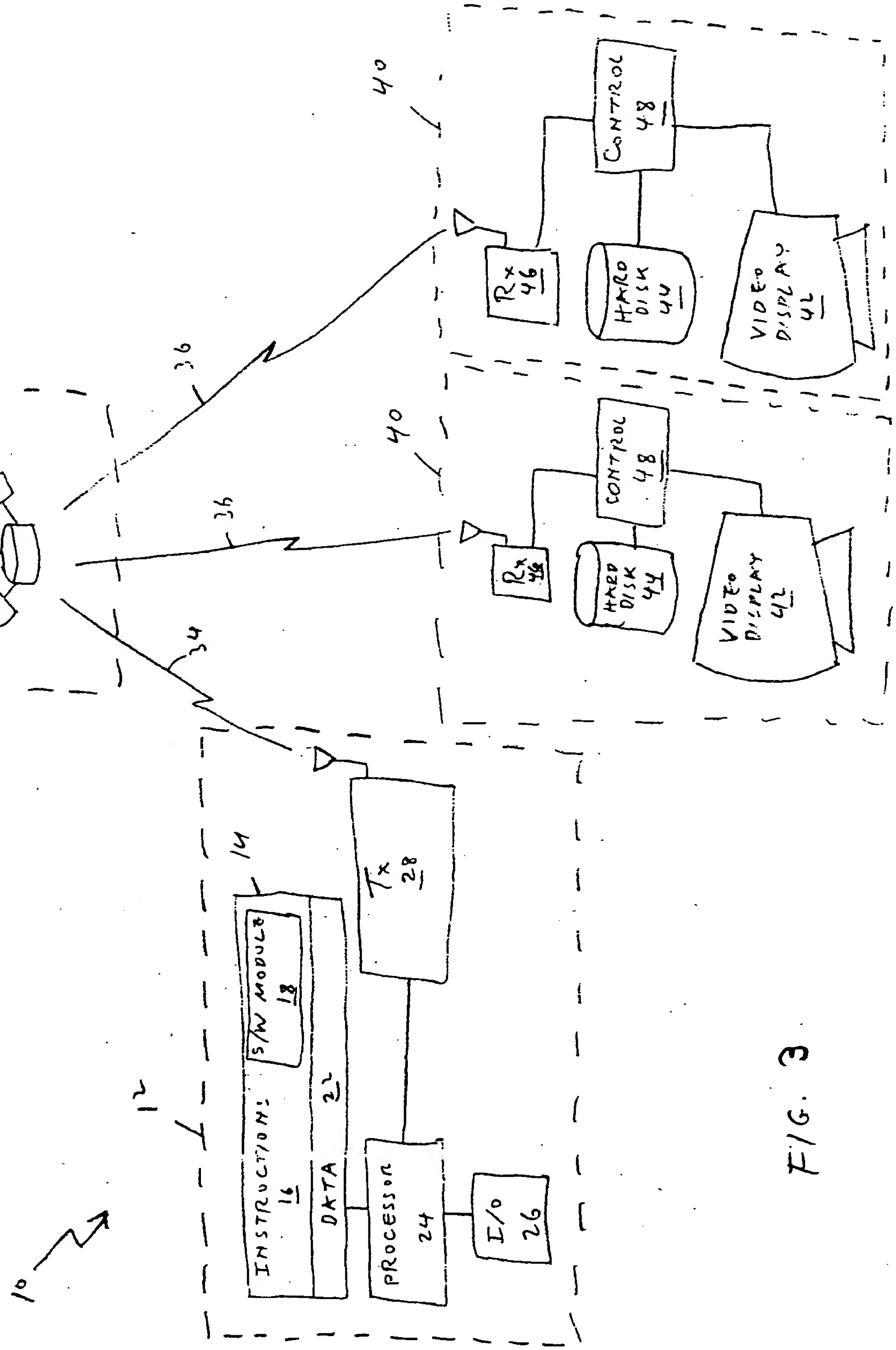

10
12
14
INSTRUCTIONS 16
S/W MODULE 18
DATA 22
PROCESSOR 24
I/O 26
Tx 28
34
20
32
36
36
40
40
Rx 46
HARD DISK 44
VIDEO DISPLAY 42
CONTROL 48

FIG. 3